



**HAL**  
open science

## Modélisation géométrique par croquis

Cédric Bobenrieth

► **To cite this version:**

Cédric Bobenrieth. Modélisation géométrique par croquis. Modélisation et simulation. Université de Strasbourg, 2019. Français. NNT : 2019STRAD047 . tel-02528289

**HAL Id: tel-02528289**

**<https://theses.hal.science/tel-02528289>**

Submitted on 1 Apr 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.

*ÉCOLE DOCTORALE DE MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE  
L'INGÉNIEUR*

Laboratoire ICube, Equipe IGG

Thèse de doctorat présentée par :

**Cédric BOBENRIETH**

soutenue le : 4 décembre 2019

pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/Spécialité : **Informatique**

---

## Modélisation géométrique par croquis

---

**Thèse dirigée par :**

Hyewon SEO

Frédéric CORDIER

Chargée de recherche, Université de Strasbourg

Maître de conférences, UHA

**Thèse encadrée par :**

Arash HABIBI

Maître de conférences, Université de Strasbourg

**Rapporteurs :**

Stefanie HAHMANN

Loïc BARTHE

Professeure des universités, Grenoble INP

Professeur des universités, Université de Toulouse

**Examineurs :**

Christian GENTIL

Professeur des universités, Université de Bourgogne

# Résumé

De nos jours, la modélisation 3D est omniprésente, cependant les outils modernes pour créer des modèles 3D sont complexes et requièrent beaucoup de temps. A contrario, l'esquisse est un moyen naturel de communiquer rapidement des idées, ainsi une méthode permettant la reconstruction automatique d'objets 3D à partir d'un croquis simplifierait ce processus. Cette méthode devrait résoudre deux problèmes : le calcul des parties cachées de la forme dessinée et la détermination des coordonnées 3D à partir des données 2D du croquis. Dans cette thèse, nous présentons deux nouvelles approches qui visent à surmonter ces problèmes. La première se sert d'a priori et d'une base de données préexistantes pour permettre la reconstruction 3D automatique de fleurs à partir d'un seul croquis selon n'importe quel angle de vue. La seconde permet la reconstruction de tout type d'objets, sans limitations, en utilisant un style de dessin plus informatif et en étant guidée par l'utilisateur.

# Remerciements

Je tiens à remercier grandement mes encadrants Hyewon Seo, Frédéric Cordier et Arash Habibi de m'avoir guidé et accompagné tout au long de ce travail de thèse au sein de l'équipe IGG du laboratoire ICube. Je suis ravi d'avoir travaillé et appris aux côtés de ces trois personnes qui, apportant chacune leurs compétences et visions des problèmes, m'ont grandement conseillé au cours de l'élaboration de cette thèse.

Je voudrais ensuite remercier Dominique Bechmann qui a énormément oeuvré afin de rendre cette thèse possible.

J'exprime toute ma gratitude aux chercheurs qui ont accepté de prendre le rôle de rapporteurs de mon mémoire : Stefanie Hahmann et Loïc Barthe.

Je remercie également Christian Gentil d'avoir accepté de faire partie de mon jury de thèse. Je témoigne également toute ma reconnaissance à David Braun, mon collègue de bureau, qui a joué pour moi un rôle de mentor tout au long de cette thèse.

Pour finir, je voudrais remercier mes amis et ma famille, et plus particulièrement ma femme et mon frère, pour leur soutien et leurs encouragements dans tout ce que j'ai entrepris.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>14</b>
1.1	La problématique de la modélisation 3D et les motivations pour un outil basé sur les croquis . . . . .	14
1.2	Les difficultés de la modélisation géométrique par croquis . . . . .	15
1.3	Les différentes techniques de modélisation par croquis . . . . .	16
1.4	Vue d'ensemble de nos contributions . . . . .	18
1.4.1	Reconstruction à l'aide de connaissances : les fleurs . . . . .	18
1.4.2	Reconstruction de formes 3D libres . . . . .	19
<b>2</b>	<b>État de l'Art</b>	<b>20</b>
2.1	Reconstruction par gestuelle . . . . .	20
2.2	Reconstruction de reliefs . . . . .	24
2.3	Modification de la posture d'un modèle 3D . . . . .	26
2.4	Reconstruction à partir de croquis sous de multiples vues . . . . .	29
2.5	Reconstruction de surface à partir d'un réseau de courbes 3D . . . . .	32
2.6	Reconstruction guidée par des données . . . . .	34
2.6.1	Connaissances a priori . . . . .	34
2.6.2	Composition de modèles préexistants . . . . .	41
2.6.3	Utilisation de modèles 3D comme support . . . . .	44
2.7	Réseaux de neurones . . . . .	47
2.8	Récapitulatif . . . . .	52
2.9	Nos travaux . . . . .	54
<b>3</b>	<b>Reconstruction 3D de Fleurs à partir de croquis</b>	<b>56</b>
3.1	Introduction . . . . .	56
3.1.1	Problématique . . . . .	57
3.1.2	Reconstruction de croquis guidée par des données 3D . . . . .	58
3.1.3	Modeleurs floraux préexistants . . . . .	60
3.1.4	Contribution . . . . .	61
3.2	Aperçu . . . . .	61
3.3	Construction d'une base de données de modèles floraux . . . . .	63
3.4	Identification des éléments botaniques basée sur l'esquisse . . . . .	65
3.4.1	Segmentation du dessin . . . . .	65

3.4.2	Identification des éléments botaniques . . . . .	68
3.5	Reconstruction du cône 3D . . . . .	73
3.5.1	Orientation de la fleur . . . . .	74
3.5.2	Orientation de la corolle . . . . .	76
3.6	Création du modèle 3D par assemblage des éléments botaniques . . . . .	78
3.6.1	Placement des pétales . . . . .	78
3.6.2	Positionnement de la tige et des feuilles . . . . .	80
3.6.3	Raffinement du maillage des pétales . . . . .	81
3.7	Résultats . . . . .	85
3.7.1	Limitations . . . . .	88
3.8	Conclusion . . . . .	88
<b>4</b>	<b>Descriptive : L'utilisation d'un dessin en style descriptif pour la mo-</b>	
	<b>délisation géométrique de formes libres</b>	<b>92</b>
4.1	Introduction . . . . .	92
4.1.1	Problématique . . . . .	93
4.1.2	Travaux connexes . . . . .	95
4.1.3	Contribution . . . . .	97
4.2	Interactions avec l'utilisateur . . . . .	97
4.3	Génération de surface à partir des courbes 3D . . . . .	101
4.4	Reconstruction 3D des courbes fermées 2D . . . . .	104
4.4.1	Fonction objective de planarité . . . . .	104
4.4.2	Spécifieurs locaux . . . . .	109
4.4.3	Ordonnancement en profondeur des segments de courbe 2D . . . . .	112
4.4.4	Résoudre le problème d'optimisation . . . . .	116
4.5	Étude utilisateur . . . . .	117
4.6	Résultats et discussion . . . . .	123
4.6.1	Comparaison avec les travaux précédents . . . . .	124
4.6.2	Limitations . . . . .	128
4.7	Conclusion . . . . .	128
<b>5</b>	<b>Conclusion</b>	<b>131</b>
5.1	Discussion . . . . .	131
5.2	Perspective de recherche . . . . .	133
<b>A</b>	<b>Autres reconstructions de fleurs</b>	<b>134</b>
<b>B</b>	<b>Exemples d'apprentissage</b>	<b>135</b>
<b>C</b>	<b>Résultats obtenus lors de l'étude utilisateur</b>	<b>136</b>
<b>D</b>	<b>Modèles supplémentaires produits avec Descriptive</b>	<b>139</b>

# Table des figures

1.1	L'interface graphique du logiciel de modélisation 3DS Max 2019. . . . .	15
1.2	Pour un même dessin 2D (a) une infinité de modèles 3D correspondants existent (b). . . . .	16
1.3	Aperçu de notre méthode permettant la reconstruction de fleurs à partir d'un croquis. . . . .	18
1.4	Aperçu de la reconstruction de formes libres avec notre seconde méthode. Une forme imaginaire d'un arbre avec une base de verre en (a) et une maison en (b). . . . .	19
2.1	Une partie de la gestuelle employée dans SKETCH [ZHH96]. . . . .	21
2.2	L'ensemble des primitives géométriques utilisées par <i>GIDeS</i> [PJBF00]. . . . .	22
2.3	Le processus de création par annotations de la méthode <i>Structured Annotations for 2D-to-3D modeling</i> [GIZ09]. . . . .	22
2.4	Résultats obtenus par la méthode BendSketch (fig. 1 [LPL <sup>+</sup> 17]). . . . .	23
2.5	À l'aide d'une seule image et des annotations données par l'utilisateur, la méthode <i>Interactive High-Relief Reconstruction for Organic and Double-Sided Objects from a Photo</i> [YHJ <sup>+</sup> 17] est capable de produire des hauts-reliefs. . . . .	25
2.6	Comparaison entre les résultats obtenus par <i>Ink-and-Ray</i> [SKv <sup>+</sup> 14] sur la ligne du haut avec ceux de <i>Seamless Reconstruction of Part-Based High-Relief Models from Hand-Drawn Images</i> [DNJ <sup>+</sup> 18] ligne du bas. Les flèches pointent sur les différences de lissage entre les deux méthodes. . . . .	25
2.7	En utilisant comme données d'entrée un croquis représentant la pose souhaitée et le modèle 3D articulé correspondant, <i>Gesture3D</i> [BVS16] effectue le repositionnement automatique du personnage. . . . .	26
2.8	Plusieurs exemples de repositionnement effectués avec les applications de <i>Sketch Abstractions for Character Posing</i> [HMC <sup>+</sup> 15] : (a) déformation d'un visage en utilisant la troisième application en dessinant par-dessus le modèle 3D, (b) repositionnement complet d'un personnage avec un dessin de contours ou une représentation en bâtonnets. . . . .	27

2.9	Résultats produits par les méthodes qui utilisent des lignes d'action pour effectuer le repositionnement de modèles 3D : (a) <i>The Line of Action : an Intuitive Interface for Expressive Character Posing</i> [GCR13] permet de poser un personnage en une seule ligne, la représentation d'un déplacement que doit suivre un caractère est effectuée avec <i>SketchiMo : Sketch-based Motion Editing for Articulated Characters</i> [CiRL <sup>+</sup> 16] par le biais de courbes (b) et <i>Differential Blending for Expressive Sketch-Based Posing</i> [OBP <sup>+</sup> 13] utilisé pour repositionner un os préalablement sélectionné par l'utilisateur en un seul tracé(c). . . . .	29
2.10	Le modèle 3D est généré automatiquement par <i>3D modeling with silhouettes</i> [RDI10] en calculant l'intersection des silhouettes fournies par l'utilisateur. . . . .	30
2.11	Processus de modélisation avec <i>Progressive sketching with instant previewing</i> [WZS19] en dessinant successivement des coupes transversales sur les trois plans orthogonaux. . . . .	31
2.12	Résultats produits avec <i>Fibermesh</i> [NISHA07] sur la première ligne (a), et formes simples créées avec <i>ShapeShop</i> [SWSJ07] seconde ligne (b). . . .	31
2.13	Création d'un échafaudage permettant à l'utilisateur de dessiner un cube en 3D avec la méthode <i>Analytic Drawing of 3D Scaffolds</i> [SKSK09]. . . . .	33
2.14	À partir du dessin en entrée, <i>True2Form</i> [XCS <sup>+</sup> 14] détecte automatiquement les indices de régularité géométrique dans le dessin de manière à produire le modèle 3D. . . . .	34
2.15	Exemple de modèles produits par <i>Modeling 3D animals from a side-view sketch</i> [EBC <sup>+</sup> 15]. . . . .	35
2.16	Résultats obtenus par les méthodes <i>SmoothSketch</i> [KH06] (a) et <i>Free-Form Sketching of Self-Occluding Objects</i> [CSPyN11] (b). Bien que les reconstructions soient conformes au dessin, elles manquent encore un peu de profondeur et donnent une impression d'aplatissement. . . . .	37
2.17	Avec le système <i>Floral diagrams and inflorescences</i> [IOOI05] l'utilisateur fournit les informations sur la structure de la fleur à partir d'un diagramme floral (a) et en dessinant l'inflorescence qu'elle doit avoir(b). Il crée ensuite les modèles géométriques (c) utilisés pour constituer la fleur (d) qui sont alors placés pour former le modèle final (e). . . . .	37
2.18	Processus de création du maillage d'un personnage avec , <i>Automatic Single-View Character Model Reconstruction</i> de Buchanan et al. [BMD13]. À partir du dessin en entrée (a) le système calcule un squelette (b) qui est utilisé pour produire le modèle final (c),(d),(e). . . . .	38
2.19	Deux exemples de modèles organiques reconstruits automatiquement à partir d'une seule image avec la méthode <i>Automatic structuring of organic shapes from a single drawing</i> . [EPB <sup>+</sup> 19]. . . . .	39

2.20	Processus simplifié de la méthode <i>Efficient Sketch-Based Creation of Detailed Character Models through Data-Driven Mesh Déformations</i> [KYY <sup>+</sup> 15]. Un modèle référent est recherché dans la base de données à partir du dessin, et celui-ci est ensuite déformé pour correspondre au croquis. . . . .	40
2.21	Exemples de rétroprojection sous d'autres angles de vues obtenues à l'aide de <i>Model-driven Sketch Reconstruction with Structure-oriented Retrieval</i> [LHZ <sup>+</sup> 16]. . . . .	40
2.22	Avec <i>Geosemantic Snapping for Sketch-Based Modeling</i> [SAG <sup>+</sup> 13] l'utilisateur sélectionne et dispose des primitives géométriques sur le croquis. Celles-ci sont placées de manière à correspondre aux contours dessinés et à s'assembler ensemble. . . . .	41
2.23	Processus de génération de <i>Interactive sketching of urban procedural models</i> [NGDA <sup>+</sup> 16]. À partir du dessin des éléments du bâtiment fournis par l'utilisateur (a) le système recherche, dans sa base, les morceaux les plus proches (b) et s'en sert pour générer le bâtiment (c). . . . .	42
2.24	Exemples de modèles composés à l'aide de différentes méthodes : (a) une chaise réalisée avec <i>Sketch-to-Design</i> [XXM <sup>+</sup> 13] , (b) une chaise conçue avec <i>Modeling By Example</i> [FKS <sup>+</sup> 04] et (c) un pégase ainsi qu'une salle bureautique créés avec <i>Sketch-Based Search and Composition of 3D Models</i> [LF08]. . . . .	43
2.25	La composition de scènes avec différentes méthodes : (a) salle bureautique conçue avec <i>MagicCanvas</i> [SI07], (b) 2 scènes modélisées avec <i>Sketch2Scene</i> [XCF <sup>+</sup> 13]. . . . .	44
2.26	Modèles créés par des utilisateurs à l'aide de <i>SecondSkin</i> [PS15]. . . . .	45
2.27	<i>Skippy</i> [KYC <sup>+</sup> 17] permet à l'utilisateur de créer facilement une courbe 3D en s'appuyant sur un modèle préalablement chargé. . . . .	45
2.28	Utilisation de données brutes comme support pour la reconstruction. Un nuage de points obtenus à partir de multiples photos est utilisé pour la reconstruction de bâtiments avec <i>Sketch-based guided modeling of 3D buildings from oriented photos</i> [SKMW17](a), des données RGB-D sont utilisées pour définir le contexte dans lequel l'utilisateur peut créer des surfaces 3D avec <i>SweepCanvas</i> [LLZ <sup>+</sup> 17] (b). . . . .	47
2.29	Deux méthodes utilisant un squelette 3D avec la pose correspondante à celle du personnage dessiné comme données d'entrée : (a) <i>ArtiSketch</i> [LG13] qui utilise plusieurs vues de la silhouette et (b) <i>Modeling Character Canvases from Cartoon Drawings</i> [BCV <sup>+</sup> 15] qui fonctionne avec une seule. . . . .	48
2.30	Visages caricaturaux créer à partir d'un croquis à l'aide du réseau de neurones <i>DeepSketch2Face</i> [HGY17]. . . . .	49
2.31	Quelques résultats obtenus par <i>Shape Synthesis from Sketches via Procedural Models and Convolutional Networks</i> [HKYM17] le réseau de neurones spécialisé dans les modèles procéduraux. . . . .	49

2.32	<i>AtlasNet</i> [GFK <sup>+</sup> 18] est capable de produire un résultat même lorsque le réseau n'a pas été entraîné sur le type d'objet à reconstruire (a), même si ces résultats sont de moins bonne qualité que lorsque l'entraînement contenait la catégorie d'objet (b). . . . .	50
2.33	Modèles reconstruits à partir de croquis sous 2 angles de vues à l'aide du réseau de neurones <i>3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks</i> [LGK <sup>+</sup> 17]. Le modèle orange est le modèle 3D le plus proche du dessin dans les données d'entraînement du réseau. . . . .	51
2.34	Exemple du processus de reconstruction d'un oiseau avec <i>Robust Flow-Guided Neural Prediction for Sketch-Based Freeform Surface Modeling</i> [LPL <sup>+</sup> 18]. En dessinant la forme sous un nouvel angle de vue (b), l'utilisateur apporte des informations supplémentaires qui permettent de modifier la surface générée avec le premier sketch (a) afin d'obtenir la forme finale (c)(d). . .	51
2.35	Deux scènes modélisées avec <i>What You Sketch Is What You Get</i> [DAI <sup>+</sup> 18]. Chaque modèle présent dans la scène a été reconstruit à l'aide de deux ou trois croquis montrés dans les encarts. . . . .	52
2.36	Tableau récapitulatif des données d'entrées et du type de méthode. . . . .	53
2.37	Tableau récapitulatif des résultats produits, de la variété d'objets qui peuvent être construits. L'information de temps est également indiquée. Pour les méthodes interactives et incrémentales, ce temps prend en compte l'intégralité du processus, dessin et interactions, alors que pour les méthodes automatiques il ne s'agit que du temps de reconstruction après que le dessin soit fourni. Lorsque l'information de temps n'est pas disponible, les cases sont remplies de <i>n.d.</i> . . . . .	54
3.1	Reconstruction d'un modèle de fleur à partir d'un croquis : le croquis d'entrée (a), les traits de guidage fournis par l'utilisateur qui seront utilisés pour la reconstruction du cône 3D (b), le croquis segmenté en pétales et autres éléments botaniques (c), modèle reconstruit du même point de vue que l'entrée (d) ; et d'un point de vue différent (e). . . . .	56
3.2	Notre système ne fonctionne pas avec les dessins trop abstraits (a), ou trop détaillés (b), et requiert un niveau d'abstraction modéré (c). . . . .	62
3.3	Exemples de modèles d'éléments botaniques dans notre base de données : feuilles (à gauche), tiges (au milieu) et pétales (à droite). Les zones d'attachement sont indiquées en rouge et les extrémités des pétales en bleu. .	64
3.4	Cas atypiques de fusion ou de scission d'un trait : deux traits représentent la courbure d'un pétale pouvant former un ou plusieurs points de jonction (a, b), la torsion d'un pétale peut être dessinée en un trait (rouge) ou en deux (vert) (c), un seul trait doit être segmenté en plusieurs courbes représentant chacune un pétale (d). . . . .	66

3.5	Exemple de croquis qui ne peut pas être segmenté automatiquement (a). L'utilisateur est invité à dessiner un pétale référent(b), qui permet au système de récupérer la forme 3D correspondante. Ensuite, l'utilisateur localise les extrémités des pétales dans l'esquisse (points rouges (c)). Ces extrémités, ainsi que la pointe du cône (point bleu (c)), servent à placer les pétales sur le dessin (représentés en fil de fer de couleurs différentes). . . . .	67
3.6	Exemple de la génération de 4 rendus d'un pétale 3D selon 4 points de vue placés sur la sphère de vue (point rouge). . . . .	70
3.7	Exemple où l'apparence de l'élément botanique seule ne peut permettre de déduire son type. L'élément segmenté en (a) représente dans un cas une tige (b), et dans l'autre une feuille (c). De ce fait, le critère de similarité seule ne suffit pas à s'assurer du bon choix du modèle 3D. . . . .	71
3.8	Pour chaque courbe représentant un élément botanique, nous récupérons les modèles 3D ordonnés selon leurs scores de similarité et sélectionnons le modèle avec le score d'évaluation le plus élevé combinant la cohérence avec l'angle de vue du dessin, la fréquence de présence d'une même espèce de fleur et d'un même modèle 3D. . . . .	73
3.9	Après avoir tracé l'esquisse florale (a), l'utilisateur localise le point <i>centre de fleur</i> (éventuellement caché) $C_2$ où la tige se joint à la corolle (point rouge) et trace les <i>courbes d'extrémité</i> (courbes rouges) pour chaque couche de pétales (b). À partir de ces traits de guidage, interprétés comme des ellipses 2D, nous reconstruisons les cônes 3D que nous utiliserons pour guider le positionnement des éléments botaniques 3D (pétales et tige). . . . .	74
3.10	Deux cônes 3D identiques, mais de sens opposés possèdent la même projection 2D. . . . .	75
3.11	La fleur est orientée vers l'observateur si la courbe d'extrémité $E_2$ est fermée (a) ou si la tige (b) ou le segment $[S_2, O_2]$ (c) intersecte avec l'une d'entre elles. Sinon, la fleur est orientée de dos par rapport à l'observateur (d). . . . .	76
3.12	Un disque 3D $E_3$ (la base d'un cône 3D) est projeté sur le plan d'esquisse sous la forme d'une ellipse $E_2$ . . . . .	77
3.13	Placement des pétales autour d'un cône 3D. . . . .	79
3.14	(a) montre la mise en correspondance entre $S_{3D}$ , la silhouette 3D du pétale, et D, la courbe dessinée à la main du pétale par l'utilisateur, en se servant de S, la projection sur le plan d'esquisse de $S_{3D}$ . $P_t^S, P_b^S, P_t^D$ et $P_b^D$ sont les points caractéristiques de S et D respectivement. Les deux segments indiqués en pointillés et les lignes continues sont échantillonnés de manière à être composés du même nombre de sommets. (b) et (c) montre un processus similaire dans le cas où la corolle complète du dessin n'est représentée que par un seul trait. Lorsque le dessin de pétale n'est pas complet, principalement car il est caché par d'autres éléments de la fleur, seule la partie visible du pétale est prise pour établir la correspondance en utilisant le seul point caractéristique que l'on a (d). . . . .	83

3.15	Pour chaque point de la silhouette du maillage 3D du pétale, une correspondance est établie avec un des points de la courbe du dessin (a). Grâce à cette correspondance, une déformation de laplacien est appliquée, permettant de raffiner le modèle pour qu'il corresponde parfaitement à l'attente de l'utilisateur (b). . . . .	85
3.16	L'interface utilisateur est composée de deux fenêtres. La première montre l'esquisse de l'utilisateur avec le centre de fleur (la croix rose) indiquée par ce dernier. On peut également y voir les trois ellipses calculées à partir des courbes d'extrémités fournies par l'utilisateur (a). La deuxième fenêtre (b) montre le résultat de la reconstruction, le modèle final 3D, dans un rendu en fil de fer, et propose également une visualisation des cônes 3D déduits à partir des ellipses (la base des cônes étant les cercles partiellement remplis en vert). . . . .	86
3.17	Différents résultats obtenus avec notre modeleur. De gauche à droite : le croquis d'entrée réalisé par l'utilisateur ; le modèle 3D reconstruit en assemblant les différents éléments floraux récupérés dans la base de données uniquement ; le modèle final avec la déformation appliquée selon le même point de vue que sur le dessin ; le modèle final déformé vu d'une direction différente et le modèle avec une texture appliquée manuellement. . . . .	89
3.18	Lorsque le dessin est dessiné selon une vue strictement de profil (a), les courbes des pétales segmentées ne retranscrivent pas la courbure nécessaire au modèle 3D. L'étape de récupération d'éléments floraux dans la base de données peut alors retourner des pétales correspondant parfaitement à la silhouette, mais complètement plats (b), alors que l'utilisateur désirerait des pétales avec une certaine courbure (c). . . . .	90
4.1	Vue d'ensemble du processus de reconstruction avec notre modeleur. Le dessin fourni par l'utilisateur en style descriptif(a), l'utilisateur assemble des segments de courbes pour générer les courbes fermées correspondant aux limites des éléments fermés composant le modèle final (b). La position 3D des points de la courbe sont calculées (c) et la surface 3D est générée (d). . . . .	92
4.2	Exemple de dessin de concepteur. Les éléments sont représentés par un ensemble de courbes dont le but est de transcrire parfaitement les propriétés géométriques de l'objet en question. Image de Alex Olieman, The Industrial Design Engineering Wiki. . . . .	96

4.3	L'utilisateur fournit un dessin initial. Les contours des parties cachées sont dessinés en pointillés (a) ; puis les segments de courbe du dessin sont décomposés en un ensemble de courbes fermées ( $C_{2D,1}$ à $C_{2D,5}$ ) (b) ; l'indice de profondeur est calculé sur chaque section des courbes fermées (c) ; un zoom de la nageoire pelvienne pour afficher les indices de profondeur (c) ; des contraintes de profondeur facultatives (indiquées par des points rouges) peuvent être définies par l'utilisateur sur certains points sélectionnés sur la courbe ; des surfaces volumiques fermées sont ensuite créées en gonflant les patchs de surface générés à partir des courbes fermées 3D(e), dont l'épaisseur du volume le long de l'axe z peut être ajustée par l'utilisateur (f). . . . .	98
4.4	Le croquis d'entrée (a) ; les courbes des éléments cachés se voient attribuer par défaut un indice de profondeur 1 (b) ; le résultat de la reconstruction montre que les parties de $C_{2D,2}$ et $C_{2D,3}$ se croisent puisqu'elles ont le même indice de profondeur (c) ; l'utilisateur peut attribuer manuellement un indice de profondeur de manière à placer $C_{2D,2}$ en dessous de $C_{2D,3}$ (d) ; le résultat de la reconstruction correspondante (e). . . . .	100
4.5	Pour chaque courbe 3D $C_{3D,i}$ (a), un patch de surface est généré par interpolation (b). En (c), ce patch de surface est dupliqué ( $M_{3D,fr}$ et $M_{3D,bk}$ ), et chacun de ces patchs est gonflé pour générer une surface volumétrique. L'épaisseur de ces surfaces est définie par des coefficients sur certains points du patch. Les coefficients de $V_{fr,5}$ , $V_{fr,8}$ et $V_{bk,6}$ ont été définis à 1,0, -0,03 et 1,0 respectivement. . . . .	102
4.6	Dans un premier temps, le triangle $T_j$ contenant $V_j$ est d'abord identifié (a). Ensuite, l'épaisseur $d_{V,j}$ en $V_j$ est calculée à l'aide du centre $O_j$ et du rayon $r_j$ du cercle circonscrit de $T_j$ . Ce cercle circonscrit est représenté en bleu clair. . . . .	104
4.7	Les matrices $R_{i:i+3}$ sont calculées dans l'équation (2) pour tous les points de $C_{3D,1}$ et $C_{3D,2}$ . $C_{3D,1}$ et $C_{3D,2}$ ont les sommets $P_1$ et $P_7$ en commun. . . . .	107
4.8	Utilisation de différentes valeurs pour le poids de la planarité de chaque courbe. Les courbes dans le plan d'esquisse (a), la reconstruction avec $\alpha_1 = 1$ et $\alpha_2 = 10$ (b), la reconstruction des mêmes courbes, mais avec $\alpha_1 = 10$ et $\alpha_2 = 1$ (c). . . . .	108
4.9	La modélisation d'un tore est effectuée en le décomposant en deux courbes fermées se chevauchant $\mathbf{C}_{3D,1}$ et $\mathbf{C}_{3D,2}$ . . . . .	108
4.10	La modélisation d'un toit (b) se fait en utilisant deux courbes (a). Ces deux courbes partagent un ensemble de points en commun $P_i, P_{i+1}$ et $P_{i+2}$ , qui sont alignés, formant ainsi le faîtage du toit 3D. . . . .	109

4.11	La courbe $C_{3D,i}$ positionnée en 3D avec des contraintes de position placées par l'utilisateur, en rouge, comme seul spécifieur utilisé(a), et les matrices calculées correspondantes $R_{i-3:i}, R_{i-2:i+1}, R_{i-1:i+2}$ et $R_{i:i+3}$ passant par le point $P_i$ qui est une contrainte de position. La même courbe $C_{3D,i}$ positionnée en 3D en plaçant les points d'angles aux points $P_i, P_{i+8}, P_{i+12}$ et $P_{i-4}$ , en plus des contraintes de position initialement placées (symbolisés par l'ajout de la couleur verte) (c), impliquant que les seules matrices calculées pour le point $P_i$ sont les matrices $R_{i-3:i}$ et $R_{i:i+3}$ . . . . .	110
4.12	La courbe $C_{2D,i}$ (a), la courbe $C_{3D,i}$ correspondante calculée en utilisant uniquement la règle de planarité (b), et la même courbe $C_{3D,i}$ calculée en spécifiant des contraintes de position aux points $P_i, P_{i+4}, P_{i+8}, P_{i+12}, P_{i+16}$ et $P_{i-4}$ (c). . . . .	111
4.13	Méthode de calcul des indices de profondeur lorsque deux courbes se chevauchent (a). Exemple où cette règle permet de déduire l'ordonnancement entre deux éléments appartenant à un chevauchement de 3 éléments (b). Initialement les traits pleins ont un indice de 0 et les pointillés de 1. Une fois que le programme détecte la paire de jonctions en T entre les deux éléments $C_{2D,i}$ et $C_{2D,j}$ , il met à jour la valeur de l'indice de profondeur du segment $C_{2D,j}$ délimité par $[P_{T,1}, P_{T,2}]$ comme étant la valeur maximale des indices de $C_{2D,i}$ au niveau de ces jonctions plus 1, c'est-à-dire dans ce cas à 2. . . . .	113
4.14	Calcul des indices de profondeur aux courbes connectées à une jonction cuspidée au point $P_C$ (a); les courbes cachées sont tracées en pointillés. Exemple d'un dessin composé de 2 courbes $C_{2D,1}$ et $C_{2D,2}$ avec une jonction en T au point $P_T$ , une jonction cuspidée en $P_C$ et deux jonctions en Y aux points $P_{Y,1}$ et $P_{Y,2}$ (b). . . . .	114
4.15	Les courbes $C_{2D,1}$ et $C_{2D,2}$ avec une jonction en T au point $P_T$ et une jonction cuspidée au point $P_C$ (a). Les contraintes d'inégalité que l'on en déduit pour les points $p_i$ et $p_j$ appartenant à la courbe $C_{3D,i}$ (b). . . . .	116
4.16	Durée de la session d'entraînement pour chaque participant. . . . .	119
4.17	Le meilleur résultat obtenu par les utilisateurs. De gauche à droite : les modèles référents, l'esquisse fournie par l'utilisateur avec les points d'angle (en vert) et les contraintes de position (en rouge), ainsi que les modèles reconstruits. . . . .	120
4.18	Temps requis pour chaque participant pour effectuer la reconstruction du modèle 3D référent correspondant. . . . .	121
4.19	Note attribuée par les participants aux questions de notre évaluation après avoir effectué leurs reconstructions. . . . .	122
4.20	Note attribuée par les participants avec une expérience antérieure avec des logiciels de modélisation 3D aux trois questions supplémentaires. . . . .	122

4.21	Ensemble de résultats obtenus avec notre système. Les contraintes spécifiées par l'utilisateur sont représentées en rouge pour les contraintes de position, en vert pour les points d'angle et vert-rouge pour les deux à la fois. . . . .	125
4.22	Comparaison avec les autres méthodes. Les cases sont cochées quand la méthode est capable de reconstruire le modèle correspondant, en supposant qu'elle ait les données adéquates en entrée. . . . .	126
4.23	La même esquisse (a) peut donner différentes formes en fonction de la manière dont l'utilisateur définit les courbes masquées et leur indice de profondeur (b). Les jonctions en T, les jonctions cuspidées et les jonctions en Y sont représentées en brun, rouge et violet respectivement. Les résultats de la reconstruction sont présentés en vue de face (c) et en perspective (d). La décomposition peut ne pas toujours être simple et peut nécessiter des informations sur le modélisateur. . . . .	129
A.1	Résultats supplémentaires obtenus avec notre système dédié à la reconstruction de fleurs. . . . .	134
B.1	Les trois exemples d'apprentissage montrés à l'utilisateur durant la session d'entraînement de l'étude utilisateur de notre modèleur <i>Descriptive</i> . De haut en bas : le modèle à reconstruire, le dessin fournit en entrée avec les spécifieurs (rouges contraintes de positions, vert points d'angle) et enfin le modèle final reconstruit. . . . .	135
C.1	Résultats obtenus lors du premier essai de nos utilisateurs pour la reconstruction de la fenêtre. Pour chaque participant : son dessin, avec les spécifieurs, ainsi que le modèle reconstruit. . . . .	136
C.2	Résultats obtenus lors du premier essai de nos utilisateurs pour la reconstruction du pantalon. Pour chaque participant : son dessin, avec les spécifieurs, ainsi que le modèle reconstruit. . . . .	137
C.3	Résultats obtenus lors du premier essai de nos utilisateurs pour la reconstruction du tyrannosaure. Pour chaque participant : son dessin, avec les spécifieurs, ainsi que le modèle reconstruit. . . . .	138
D.1	Modèles produits avec <i>Descriptive</i> . De gauche à droite : le dessin sans parties cachées, le dessin avec les parties cachées et les spécifieurs, le modèle obtenu. . . . .	140

# Chapitre 1

## Introduction

Ce mémoire de thèse présente les différents travaux de recherche effectués au sein de l'équipe IGG du laboratoire ICube, rattaché à l'Université de Strasbourg. Ce premier chapitre a pour objectif de présenter brièvement la modélisation 3D et d'expliquer pourquoi un outil de reconstruction basée sur des croquis serait utile, puis d'exposer les difficultés sur la création d'un tel outil qui sont à l'origine du sujet de recherche sur la modélisation géométrique par croquis. Enfin, il introduira les travaux effectués au cours de ma thèse et leurs contributions.

### 1.1 La problématique de la modélisation 3D et les motivations pour un outil basé sur les croquis

De nos jours, la modélisation 3D est omniprésente. Dans un premier temps utilisée principalement comme outil de conception en industrie ou en architecture, les modèles 3D ont très vite été présents dans de nombreux autres domaines tels que la conception de jeux vidéo et de mondes virtuels, qui prend de l'ampleur avec l'arrivée sur le marché des casques de réalité virtuelle, ou encore dans le monde des effets spéciaux et du cinéma. De plus, de nouvelles applications de modélisation plus abordables sont désormais présentes sur le marché grand public rendant ce domaine accessible à de plus en plus de personnes. Cependant, malgré l'expansion de l'utilisation de la modélisation 3D, et de la demande croissante en logiciel l'accompagnant, cette tâche reste trop complexe pour qu'elle soit effectivement à la portée de tous et nécessite un temps d'apprentissage qui décourage de nombreuses personnes. En effet, les logiciels spécialisés en modélisation 3D les plus communs (3DS Max, Maya, ou Blender par exemple) consistent en un ensemble vaste d'outils 3D complexes (fig. 1.1) pour les non-initiés, et qui ne sont pas naturels pour les artistes [Mar03].

La création d'un modèle 3D reste un processus qui requiert du temps et des efforts, y compris pour les experts, qui vont alors préférer esquisser en quelques minutes leurs idées sur une simple feuille de papier. En effet, le dessin est un moyen naturel de communiquer rapidement des idées [OSSJ09] [CA09] et reste une des premières étapes dans

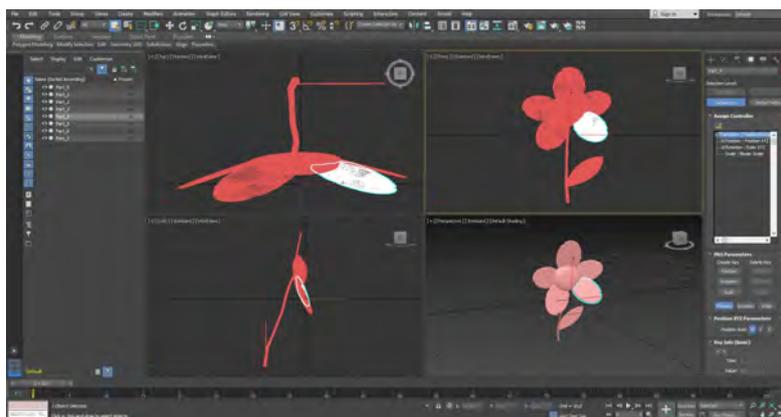


FIGURE 1.1 – L’interface graphique du logiciel de modélisation 3DS Max 2019.

les processus de conception, y compris dans la conception de modèles 3D où le croquis sert de référence. Avoir un programme capable d’interpréter automatiquement le dessin réalisé par l’artiste et d’en produire un modèle 3D serait alors un outil idéal. Puisque la modélisation elle-même commence par un prototypage en 2D, ce système permettrait d’éviter l’étape de modélisation manuelle qui requiert du temps et des efforts. Ce genre d’outil rendrait donc l’intégralité du processus de création plus rapide, donc moins coûteux, et trouverait sa place dans de nombreux domaines industriels. De plus, puisque tous les hommes ont la capacité de dessiner, il réussirait à faire ce que les logiciels de modélisation actuels ont échoué : rendre la reconstruction 3D réellement accessible à tous sans effort. C’est donc dans l’optique de concevoir un tel programme que le domaine de recherche sur la modélisation géométrique par croquis a vu le jour.

## 1.2 Les difficultés de la modélisation géométrique par croquis

Notre cerveau est facilement capable de se représenter une image mentale 3D de l’objet esquissé, ce qui implique que notre perception de ces contours et leur analyse reposent sur différentes hypothèses concernant les surfaces visualisées ([Mar77]). Lorsque nous analysons un croquis, nous commençons par examiner les courbes qui le composent. Notre cerveau va alors réussir à produire une image mentale 3D de ce qui est représenté sur le papier à l’aide de certains principes de perception que nous suivons inconsciemment. Ces principes ont été largement étudiés par les psychologues de la Gestalt-thérapie et incluent par exemple l’anticipation de la symétrie et de la simplicité ou encore l’alignement et la régularité. Entre autres, nous nous représentons également toujours un objet avec sa compacité maximale [LPS08], ainsi si nous observons, par exemple, un cercle 2D nous l’interprétons comme une sphère en 3D.

Malgré tout, réussir à interpréter algorithmiquement un dessin n’est pas une tâche

aisée. Même si l'on arrivait à automatiser l'ensemble des règles de perceptions auxquelles nous sommes soumises inconsciemment, cela ne résoudrait pas le problème fondamental au cœur de la modélisation géométrique par croquis : pour chaque courbe 2D présente sur le dessin, il existe une infinité de courbes 3D correspondantes, autrement dit il existe une infinité de formes 3D qui correspondent à la description 2D faite par les contours de l'utilisateur comme le montre la fig. 1.2.

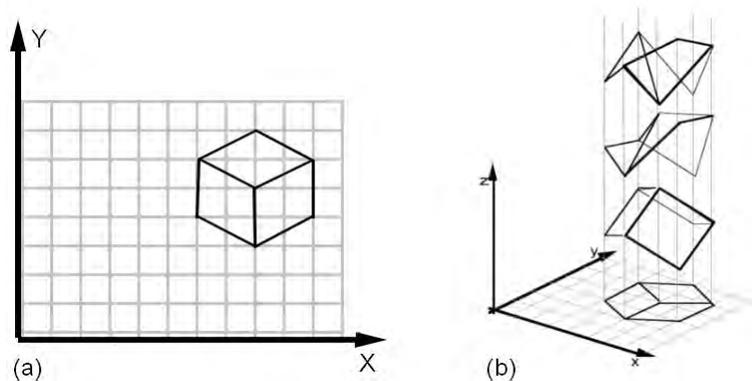


FIGURE 1.2 – Pour un même dessin 2D (a) une infinité de modèles 3D correspondants existent (b).

Une autre difficulté à laquelle les programmes doivent se heurter concerne les parties cachées. En effet, puisqu'un croquis est la représentation 2D d'une forme sous un seul angle de vue, il est très fréquent que des parties de celle-ci se retrouvent cachées par d'autres et qu'elles ne soient donc pas présentes sur le dessin ou uniquement partiellement. Il est alors nécessaire que le programme soit capable de déduire et compléter les parties manquantes de l'objet de manière à pouvoir effectuer la reconstruction 3D d'une telle forme. Cependant, même si cette complétion des parties manquantes est une tâche que notre cerveau effectue avec facilité, ce n'est pas le cas algorithmiquement. En effet, la perception que l'on a des courbes dépend des connaissances préalables que nous avons et du contexte dans lequel est effectué le croquis [Pal75] [SK08] ; c'est de cette manière que nous sommes capables d'identifier ce que représente le dessin, et ce sont nos connaissances sur ce type d'objet qui nous permette d'en déduire les parties manquantes. Un programme ne dispose pas de toutes ces données et ne peut donc comprendre la sémantique du dessin, ce qui rend la complétion automatique extrêmement compliquée.

### 1.3 Les différentes techniques de modélisation par croquis

De nombreux travaux ont été effectués par des chercheurs au cours de ces dernières décennies afin d'apporter des solutions à ce problème de reconstruction géométrique

basée sur des croquis. Les méthodes développées peuvent être regroupées dans les différentes catégories suivantes :

- *Modélisation géométrique par gestuelle et annotations* : Des méthodes portent sur l'interprétation des courbes esquissées par l'utilisateur comme commande de modélisation dans le but de reconstruire un objet 3D. L'utilisateur se sert donc d'annotations pour apporter des informations supplémentaires sur le type des courbes dessinées et aider le programme à effectuer la reconstruction.
- *Modélisation géométrique de reliefs* : Puisque la modélisation 3D à partir d'un croquis est un problème complexe, certains chercheurs ont décidé de se focaliser sur la reconstruction d'un relief permettant un rendu 3D de la forme dessinée, bien que limité au point de vue de l'esquisse.
- *Modification de la posture d'un modèle 3D* : Des méthodes ont été développées afin de créer un outil permettant la modification de la pose d'un modèle 3D, déjà existant, de manière simple et rapide en utilisant un croquis 2D.
- *Modélisation de surfaces à partir d'un réseau de courbes 3D* : Plutôt que de chercher à reconstruire un maillage, certains travaux ont porté sur la reconstruction de courbes 3D uniquement, c'est-à-dire trouver la position 3D des courbes représentées dans le croquis. Ce réseau de courbes 3D peut par la suite être utilisé pour générer des surfaces par des méthodes de reconstruction de maillage à partir de courbes 3D.
- *Modélisation à l'aide de plusieurs vues* : De manière à obtenir suffisamment d'informations pour la reconstruction 3D, une solution adoptée par plusieurs scientifiques est d'utiliser de multiples croquis représentant la même forme sous différents points de vues. De cette façon, leurs méthodes n'ont pas de problèmes d'occultation des parties cachées, puisque l'ensemble des vues permettent de décrire l'intégralité de l'objet.
- *Modélisation géométrique guidée par des données* : Une autre approche pour obtenir les informations supplémentaires à la reconstruction est d'être guidé par des données supplémentaires. Une façon d'obtenir ces informations peut être de se limiter à un type d'objet en particulier, d'utiliser des modèles 3D comme données d'entrée supplémentaires fournies par l'utilisateur, ou encore de se servir d'une base de données 3D préexistante pour trouver un modèle référent qui servira de guide à la reconstruction.
- *Reconstruction 3D par réseaux de neurones* : apparus plus récemment, des travaux ont porté sur l'utilisation de l'intelligence artificielle, à travers les réseaux de neurones, pour automatiser le processus de modélisation géométrique 3D à partir d'une esquisse.

Le chapitre section 2.9 présente de manière plus détaillée un ensemble de travaux pour chacune de ces catégories, ainsi que leurs avantages et leurs limitations.

## 1.4 Vue d'ensemble de nos contributions

Malgré les avancées technologiques et scientifiques, ce domaine de recherche n'a pas encore trouvé de solutions répondant à tous les problèmes et reste donc ouvert. Les travaux effectués au cours de cette thèse visent à développer de nouvelles approches pour permettre d'élargir la portée de reconstruction des méthodes actuelles tout en conservant la simplicité que doivent avoir les méthodes de modélisation basée sur des croquis. La première méthode que nous avons mise au point fait partie des méthodes de modélisation géométrique guidée par des données se limitant à un type d'objets, et permet la reconstruction 3D de modèles détaillés de fleurs. Elle est présentée à la section 1.4.1 et détaillée à la section 3.8. La seconde méthode, quant à elle, permet la reconstruction 3D de n'importe quelle forme, en utilisant un nombre minimal d'annotations fournies par l'utilisateur. Elle est introduite à la section 1.4.2 et détaillée à la section 4.7.

### 1.4.1 Reconstruction à l'aide de connaissances : les fleurs

L'utilisation de modèles 3D de fleurs par les botanistes est de nos jours relativement courante. De plus avec la nécessité de plus en plus grande de concevoir des mondes virtuels, que ce soit pour les jeux vidéo ou le cinéma, la capacité à produire de multiples environnements 3D est nécessaire. Pourtant la création de modèles floraux reste encore un processus demandant énormément de temps, ne serait-ce que pour la modélisation d'un seul type de fleur, et cela même avec des modeleurs spécialement dédiés à leurs reconstructions. Or la création des mondes virtuels requiert d'avoir une large diversité de flore pour offrir une expérience réaliste. Un outil qui permettrait la modélisation rapide de tels modèles est donc nécessaire.

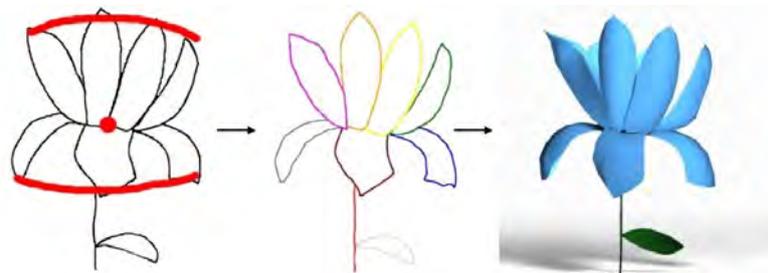


FIGURE 1.3 – Aperçu de notre méthode permettant la reconstruction de fleurs à partir d'un croquis.

Nous avons mis au point une application permettant la modélisation d'objets floraux à partir d'un croquis d'une vue représentant une seule fleur. Notre système utilise des connaissances a priori sur la structure botanique que doivent avoir les fleurs et s'appuie sur une base de données d'éléments botaniques 3D afin de reconnaître les parties dessinées par l'utilisateur et de composer le modèle 3D final. Notre système permet la production

de modèles 3D détaillés pour une large variété d'espèces florales en ne nécessitant que très peu d'interactions avec l'utilisateur, le tout en quelques minutes seulement.

### 1.4.2 Reconstruction de formes 3D libres

Notre premier système se limite à la reconstruction de fleurs. Bien qu'il permette une reconstruction de modèles détaillés, sa limitation ne le rend pas utilisable dans l'intégralité des domaines. De manière à nous rapprocher du but de la modélisation géométrique par croquis, nous avons cherché à développer un programme capable de modéliser n'importe quelle forme, sans aucune restriction sur la nature de l'objet, à partir d'un dessin d'une seule vue.

De manière à résoudre le problème d'occultation sans avoir besoin d'informations sémantiques sur la nature de l'objet dessiné, ce qui limiterait le champ des reconstructions possibles, notre méthode se sert d'un système de contraintes indiquées par l'utilisateur. Ce dernier accompagne le programme et guide la reconstruction de manière à lever les différentes ambiguïtés liées à l'utilisation d'un croquis unique. Ces interactions restent cependant extrêmement simples en comparaison des opérations qu'un artiste doit effectuer sur un modèleur 3D commercial, et permettent à notre système de reconstruire tout type d'objets qu'ils soient organiques ou artificiels.

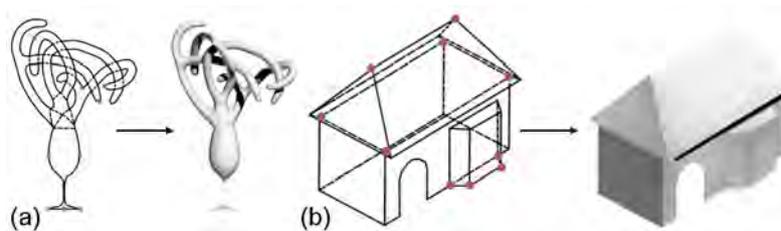


FIGURE 1.4 – Aperçu de la reconstruction de formes libres avec notre seconde méthode. Une forme imaginaire d'un arbre avec une base de verre en (a) et une maison en (b).

## Chapitre 2

# État de l'Art

La manière d'inférer un modèle 3D à partir des courbes d'une esquisse est le cœur du domaine de recherche de modélisation géométrique par croquis. La problématique majeure, inhérente à la reconstruction de formes 3D, provient du manque d'informations présentes dans un dessin 2D sur la forme générale de l'objet et de ses parties cachées ; de simples contours 2D ne donnant quasiment aucune information sur la surface 3D qu'ils sont censés décrire. De nombreux travaux ont été développés de manière à apporter une réponse à ce problème et font suite à *Teddy* de Igarashi et al [IIMT07] et *SKETCH* de Zeleznik et al. [ZHH96], souvent nommés par les études ([DL16], [CA09], [OSSJ09]) comme étant les premiers systèmes de modélisation 3D basés sur des croquis.

### 2.1 Reconstruction par gestuelle

Une première approche exploitée par plusieurs travaux, pour combler le manque d'information qu'ont les croquis 2D pour la reconstruction de modèles 3D, consiste en l'utilisation de règles de tracé spécifiques que l'utilisateur doit suivre, de manière à fournir des informations plus détaillées sur la géométrie de l'objet à reconstruire, tel que les zones convexes ou concaves, le parallélisme, etc. *SKETCH*, de Zeleznik et al., [ZHH96] a été le premier système à adopter cette approche. Il se base sur une *gestuelle* qui vise à fournir suffisamment d'informations pour sélectionner le type de primitive géométrique à instancier et déterminer ses dimensions ainsi que son positionnement en 3D. Cette *gestuelle* consiste en un ensemble de combinaisons particulières entre les trois boutons de la souris, le tracé des courbes et le clavier ; et sert à décrire l'ensemble des opérations possibles dans le système. Un exemple de quelques combinaisons est montré fig. 2.1. Cependant, cette *gestuelle* est également le point faible de leur méthode : en plus d'être limitée à peu de primitives, elle est relativement complexe et nécessite que l'utilisateur mémorise un ensemble de combinaisons différentes d'entrées (tracé, bouton souris et clavier) ainsi que leurs significations.

*GIDeS* de Pereira et al. [PJBF00] a été développé pour pallier ce problème. Ayant été conçu pour être utilisé avec une tablette, donc sans souris et sans clavier, la gestuelle ne

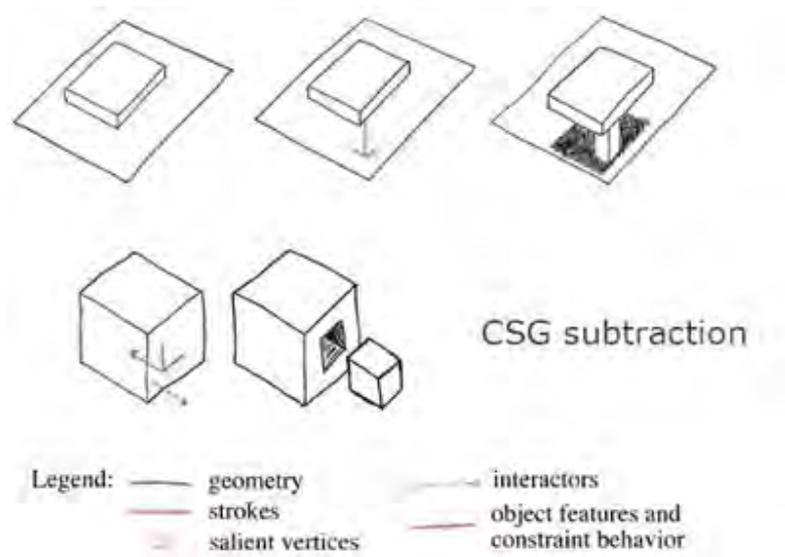


FIGURE 2.1 – Une partie de la gestuelle employée dans SKETCH [ZHH96].

doit être composée que de la seule information possible, le tracé effectué par l'utilisateur, la simplifiant comparé à *SKETCH*. De plus, un système d'icônes est mis au point de manière à résoudre le problème de la mémorisation de la gestuelle. Lorsque l'utilisateur trace un trait, le modeler va afficher une (ou plusieurs) icône(s) de manière à indiquer la façon dont il l'interprète, et les primitives qui seront créées. L'utilisateur peut alors cliquer sur l'icône correspondant à l'instruction qu'il désirait transmettre avec ce trait, ou le modifier pour obtenir l'interprétation attendue. De ce fait, l'utilisateur n'a pas besoin de mémoriser parfaitement l'intégralité des règles de tracé, puisque les icônes du système vont lui servir de repère. Cependant, tout comme *SKETCH* ces méthodes sont limitées dans les formes reconstructibles par la taille de la gestuelle et l'utilisation de primitives géométriques, montrées fig. 2.2, pour composer le modèle final.

*Structured Annotations for 2D-to-3D modeling* de Gingold et al. [GIZ09] utilise une autre approche. Au lieu de se servir d'une gestuelle qui sert d'instruction pour indiquer quelle primitive modéliser, des annotations sont ajoutées sur le dessin de manière à décrire la forme correspondant au contour dessiné. L'utilisateur esquisse les éléments composant la forme finale les uns après les autres. Lorsque le contour d'un des éléments est dessiné, un cylindre généralisé ou un ellipsoïde lui correspondant est calculé, le système fournit ensuite à l'utilisateur un ensemble de widgets lui permettant de changer différentes propriétés de l'élément 3D, comme son inclinaison ou son épaisseur, afin d'obtenir la forme désirée. Par la suite, il indique à l'aide d'annotations (sous la forme de courbes tracées avec des outils spécifiques) les relations entre chacun de ces éléments 3D, tel que le fait qu'ils doivent être connectés ou symétriques fig. 2.3. L'utilisateur peut

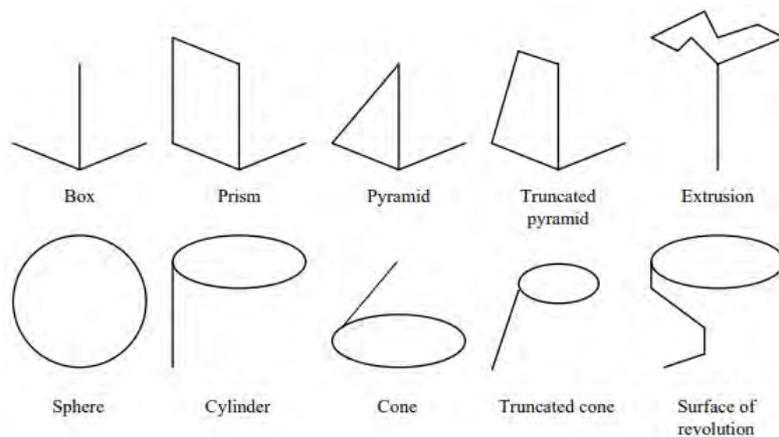


FIGURE 2.2 – L’ensemble des primitives géométriques utilisées par *GIDeS* [PJBF00].

ainsi reconstruire des modèles 3D que les méthodes précédentes ne pouvaient faire, puisqu’elle permet à l’artiste un contrôle complet sur les éléments reconstruits. Cependant, elle souffre malgré tout de certaines limitations. Tout d’abord, comme les primitives utilisées sont principalement des cylindres généralisés, elle permet surtout de reconstruire des formes organiques, et est dans l’incapacité de construire des modèles 3D avec des surfaces plates ou des arêtes saillantes. De plus, elle ne permet pas la modélisation de petits détails sur le modèle 3D. Enfin, en raison de l’utilisation de nombreux widgets 3D pour modifier la forme des éléments, leur système se trouve extrêmement proche d’un modelleur classique et un peu moins d’un véritable modelleur basé sur des croquis.

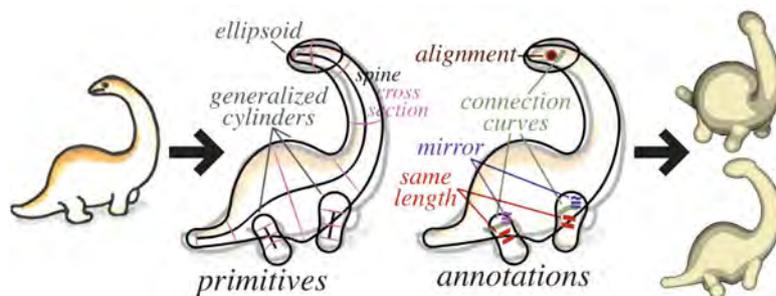


FIGURE 2.3 – Le processus de création par annotations de la méthode *Structured Annotations for 2D-to-3D modeling* [GIZ09].

Selon la même idée d’ajouter des annotations pour décrire la géométrie que doit avoir l’élément 3D dont le contour est dessiné, *BendSketch* de Li et al. [LPL<sup>+</sup>17] apporte une solution au problème du niveau de détails sur le modèle final. Leur système va

chercher à modéliser la forme 3D en utilisant des *bending lines* comme annotation, c'est-à-dire des lignes qui vont décrire la surface du maillage, tel que des zones lisses, des crêtes, des vallées, etc. Contrairement à *Structured Annotations for 2D-to-3D modeling*, *BendSketch* part du principe que la forme finale peut être décomposée en surfaces, et non pas en cylindres généralisés. Ainsi pour chaque contour fermé fourni par l'artiste, c'est une surface plane qui est reconstruite. Puis, à l'aide des annotations de l'utilisateur, indiquant les zones concaves ou convexes, un champ de direction est calculé sur l'ensemble du maillage 2D. Ce champ est ensuite utilisé pour modifier la surface et lui permettre d'obtenir sa forme finale. Par la suite, l'utilisateur peut changer l'angle de vue 3D de la surface reconstruite de manière à dessiner, et donc ajouter de manière incrémentale, une nouvelle surface composant la forme finale. Il peut également répliquer la surface existante par symétrie. En utilisant uniquement 8 types de courbes, incluant le contour du dessin, leur méthode permet de réaliser une grande variété de modèles, sans être restreint à des surfaces lisses, et pouvant présenter de nombreux détails, comme le montre la fig. 2.4 tirée de leur article.

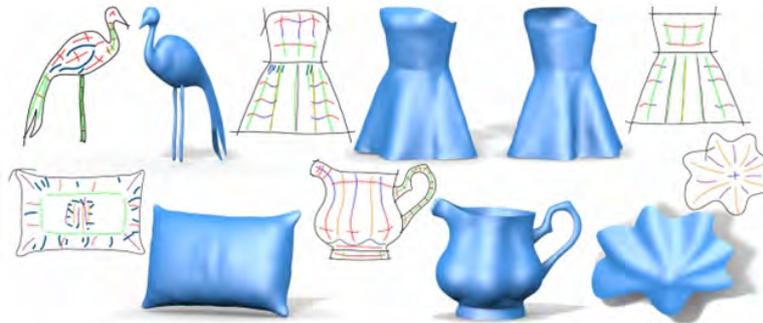


FIGURE 2.4 – Résultats obtenus par la méthode BendSketch (fig. 1 [LPL<sup>+</sup>17]).

Cette méthode a malgré tous des limites. La modélisation de formes auto-occlusives ou d'objets à couches n'est pas possible à partir d'une seule vue. Pour résoudre ce problème, ils ont intégré la capacité de dessiner de manière incrémentale, mais cela nécessite malgré tout que l'utilisateur spécifie manuellement les attachements entre chacune des surfaces. De plus, et malgré cette capacité à dessiner sous de multiples vues, la reconstruction d'un modèle très complexe (telle qu'une forme avec beaucoup d'entrelacement) reste une tâche extrêmement difficile.

Enfin, l'un des problèmes majeurs partagés par toutes les méthodes à annotations est la distance entre le dessin et la forme reconstruite. On entend par là qu'il existe une grande différence entre le dessin fourni par l'utilisateur et le modèle reconstruit dû à la présence de toutes ces annotations : le dessin se retrouve chargé avec de nombreuses courbes qui, bien qu'elles servent à donner des informations sur la forme de l'objet, ne correspondent pas forcément à une courbe réellement visible dans le modèle 3D final.

## 2.2 Reconstruction de reliefs

De manière à combler le manque d'informations sur la forme que l'objet représenté doit avoir en 3D, certains travaux ont décidé de se concentrer sur la reconstruction de reliefs plutôt que des modèles 3D complets. L'idée de base est que ces systèmes vont chercher à créer des surfaces 3D qui suffisent à effectuer un rendu 3D selon l'angle de vue du dessin, ou un angle très proche de celui-ci. *Ink-and-Ray* de Sykora et al. [SKv<sup>+</sup>14] fait partie de ces systèmes. L'utilisateur est invité à spécifier les régions correspondant aux éléments de l'objet représenté à l'aide d'annotations simples de manière à permettre au système de segmenter le croquis en contours fermés correspondant chacun à une région. Quand une de ces régions segmentées inclut des occultations, car elle est partiellement cachée par un autre élément, les contours manquants sont automatiquement estimés afin de fermer la région. Cette information sur la présence d'une partie cachée est alors utilisée, avec les annotations, pour ordonner les éléments en profondeur, c'est-à-dire déterminer quel élément est devant l'autre. Chaque région est ensuite légèrement gonflée afin de produire une surface 3D avec volume. La coloration du dessin d'origine est ensuite utilisée pour texturer les surfaces 3D, et effectuer un rendu 3D avec une illumination globale. L'un des gros défauts de cette méthode est que l'utilisateur n'a pas de contrôle direct sur la forme finale, et ne peut pas définir la profondeur que certains éléments doivent avoir, résultant en des reliefs très peu profonds (ligne du haut fig. 2.6) ; de plus elle est limitée à des formes relativement arrondies et lisses.

A contrario, *Interactive High-Relief Reconstruction for Organic and Double-Sided Objects from a Photo* de Yeh et al. [YHJ<sup>+</sup>17] est développée pour permettre la reconstruction de *hauts-reliefs*, c'est-à-dire des reliefs avec une grande profondeur permettant la visualisation du modèle reconstruit selon des angles de vues qui peuvent différer jusqu'à 45 degrés de l'angle de vue utilisé pour l'image d'origine. Bien que leur méthode ait été conçue pour fonctionner à partir de photos, l'utilisation d'un dessin est également possible (mais donnerait des résultats d'une qualité moins grande). Leur système invite l'utilisateur à piloter la reconstruction. Dans un premier temps, il indique les régions qui composent le modèle présent sur l'image, qui sont ensuite segmentées, ordonnées en profondeur et complétées. La complétion joue ici un rôle majeur puisque leur méthode détecte les cas où les deux faces d'une même surface sont exposées, lorsque l'objet est plié, donnant ainsi deux régions distinctes en 2D ; et complètent les parties cachées de manière à joindre ces deux régions en un seul élément 3D plié. L'utilisateur indique ensuite l'orientation et l'inclinaison que doit avoir chacune des régions à l'aide d'*indicateurs de pentes*. Bien qu'elle corrige le problème de profondeur d'*Ink-and-Ray*, cette méthode souffre malgré tout de la même limitation à la reconstruction d'objets organiques lisses. De plus, elle échoue lorsque l'image est trop complexe pour être segmentée, avec des détails très fins ou une forme avec beaucoup d'entrelacements.

Le travail de Dvorožňák et al. *Seamless Reconstruction of Part-Based High-Relief Models from Hand-Drawn Images* [DNJ<sup>+</sup>18] a également pour objectif de perfectionner

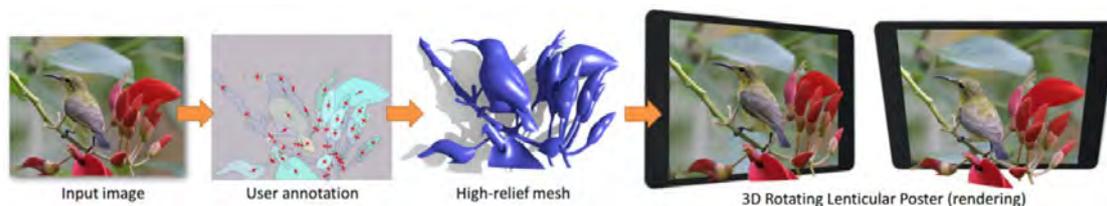


FIGURE 2.5 – À l’aide d’une seule image et des annotations données par l’utilisateur, la méthode *Interactive High-Relief Reconstruction for Organic and Double-Sided Objects from a Photo* [YHJ<sup>+</sup>17] est capable de produire des hauts-reliefs.

le travail de ces prédécesseurs. En effet les reliefs résultants de *Ink-and-Ray* présentent des coutures visibles entre chaque élément composant le résultat final, les rendant ainsi beaucoup moins naturels. Ici, les données d’entrées considérées sont les régions déjà segmentées, complétées et ordonnées en profondeur ; cette méthode vient donc se placer après les premières étapes de *Ink-and-Ray*. Contrairement à cette dernière, celle-ci va effectuer l’inflation des régions à l’aide d’un système non linéaire qui permet d’être plus raide aux limites et arrondie de manière plus uniforme, donnant ainsi un aspect plus naturel lors du rendu. Leurs résultats sont des reliefs avec un peu plus de profondeur que ceux de *Ink-and-Ray*, et plus naturel comme le montre la fig. 2.6. Cependant, elle est elle aussi limitée aux formes organiques arrondies.

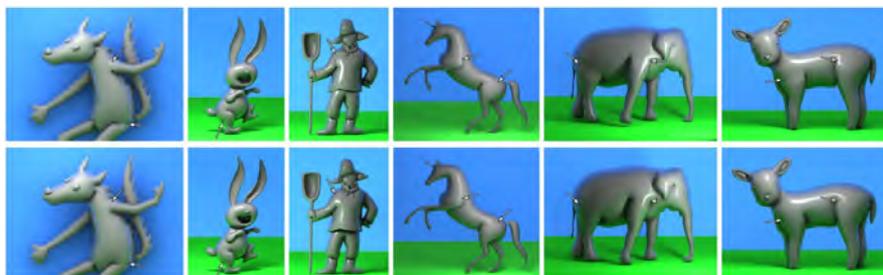


FIGURE 2.6 – Comparaison entre les résultats obtenus par *Ink-and-Ray* [SKv<sup>+</sup>14] sur la ligne du haut avec ceux de *Seamless Reconstruction of Part-Based High-Relief Models from Hand-Drawn Images* [DNJ<sup>+</sup>18] ligne du bas. Les flèches pointent sur les différences de lissage entre les deux méthodes.

L’ensemble de ces méthodes arrivent donc à produire un résultat permettant un rendu 3D à partir d’une image selon un seul point de vue. Cependant elles se trouvent restreintes dans le type d’objets, fonctionnant principalement pour des objets arrondis. De plus se limiter à la production de reliefs signifie ne pas produire de modèles 3D complets utilisables dans tous les domaines, puisque l’arrière de ces reliefs n’est pas reconstruit.

## 2.3 Modification de la posture d'un modèle 3D

Une autre approche développée par plusieurs articles consiste à utiliser le dessin non pas comme outil de modélisation, mais comme outil d'édition de modèles. Plus précisément le croquis est utilisé pour modifier la pose 3D d'un modèle articulé fourni par l'utilisateur. Seki et Igarashi [SI17] ont présentés avec *Sketch-Based 3D Hair Posing by Contour Drawing* une méthode qui se limite au repositionnement des cheveux d'un modèle 3D, fourni par l'utilisateur. Leur système va modifier les paramètres des articulations des cheveux 3D à l'aide d'un champ de directions déterminé à partir de l'esquisse de manière à ce que leurs nouvelles positions correspondent aux contours du dessin.

*Gesture3D* de Bessmeltsev et al. [BVS16] permet de positionner un personnage 3D à l'aide d'un *dessin de gestes* qui consiste en un croquis du contour du personnage relativement simple, mais très expressif, couramment utilisé par les artistes pour décrire l'action ou la posture prise par un personnage (fig. 2.7). La détection du squelette en 2D est faite de manière automatique. Par la suite, comme le modèle 3D, avec un squelette, est fourni par l'utilisateur, le système est capable de détecter la position 3D de chacun des membres du personnage en se servant de la longueur connue des os en 2D et 3D. Le système arrive alors à établir une correspondance entre le dessin et le modèle, à l'aide des positions déduites des squelettes et du fait que les contours du dessin représentent la bordure du maillage 3D. Il est alors capable de repositionner le modèle par rapport au croquis. Cependant, si le dessin est mal dessiné et que la proportion n'est pas respectée, alors le positionnement automatique risque d'échouer. De même si la position représentée est ambiguë.

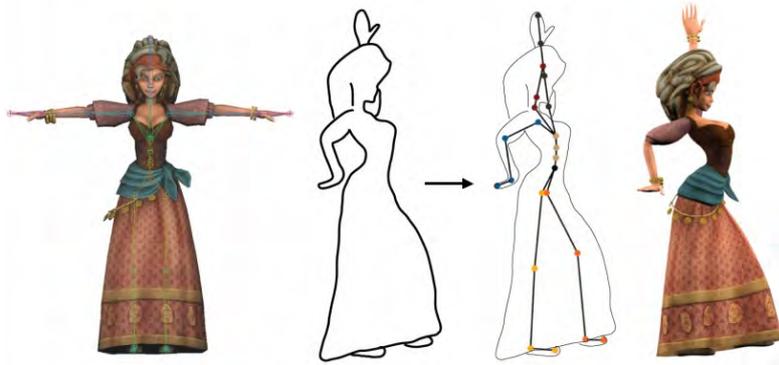


FIGURE 2.7 – En utilisant comme données d'entrée un croquis représentant la pose souhaitée et le modèle 3D articulé correspondant, *Gesture3D* [BVS16] effectue le repositionnement automatique du personnage.

*Sketch Abstractions for Character Posing* de Hahn et al. [HMC<sup>+</sup>15] propose trois applications basées sur le positionnement de modèles 3D à partir d'un croquis. La première consiste en un repositionnement complet d'un modèle à l'aide d'un dessin des contours du personnage ou en utilisant une représentation en bâtonnets du squelette de

celui-ci. La seconde permet la composition d'un nouveau personnage en utilisant une base prédéfinie d'éléments 3D découpés et du contour 2D abstrait correspondant. L'utilisateur dessine le caractère qu'il désire créer en utilisant l'ensemble de ces abstractions, et chacun des éléments 3D est alors repositionné pour correspondre au croquis puis assemblé aux autres pour former un nouveau personnage. Enfin, la troisième permet de dessiner directement sur le modèle 3D des courbes indiquant comment le maillage doit être déformé à certains endroits spécifiques. Des exemples d'utilisation de la première et de la troisième application sont montrés fig. 2.8. Leur système fonctionne en calculant la projection 2D du modèle fourni par l'utilisateur, et en cherchant ensuite les paramètres du squelette du maillage de manière à modifier sa position pour que cette projection soit le plus proche de l'esquisse de l'utilisateur. Cependant, il est limité de la même façon que *Gesture3D*, dans le cas où les proportions du dessin ne correspondraient pas à celles du modèle, il serait alors impossible d'effectuer un repositionnement. De plus, la liberté sur la position que le modèle peut prendre est restreinte par ce que permettent de faire les articulations du squelette.

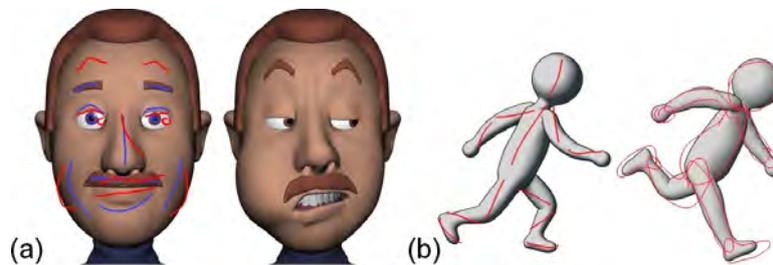


FIGURE 2.8 – Plusieurs exemples de repositionnement effectués avec les applications de *Sketch Abstractions for Character Posing* [HMC<sup>+</sup>15] : (a) déformation d'un visage en utilisant la troisième application en dessinant par-dessus le modèle 3D, (b) repositionnement complet d'un personnage avec un dessin de contours ou une représentation en bâtonnets.

De manière à simplifier au maximum ce processus de repositionnement par croquis, des travaux ce sont inspirés du monde de l'animation 2D en utilisant les *lignes d'action*. En animation 2D, ces lignes d'action sont des courbes qui capturent le mouvement et l'équilibre global du personnage. De plus, elles correspondent naturellement au squelette de celui-ci. Dans *Differential Blending for Expressive Sketch-Based Posing* de Öztireli et al. [OBP<sup>+</sup>13] l'utilisateur commence par sélectionner l'os du squelette du modèle 3D dont il désire modifier la position, et trace une ligne d'action représentant la nouvelle position désirée. Contrairement aux autres méthodes, celle-ci représente le squelette avec une structure à base de courbes. Chaque os est implicitement échantillonné en une série de nœuds consécutifs, permettant ainsi une déformation du squelette, et donc l'accès à une plus large variété de poses que les méthodes classiques, au détriment d'un temps de calcul un peu plus long (fig. 2.9 (c)).

Cette même idée d'utiliser une ligne d'action pour représenter la posture du modèle

est présente dans *The Line of Action : an Intuitive Interface for Expressive Character Posing* de Guay et al. [GCR13]. L'utilisateur définit la pose principale avec une seule ligne d'action (fig. 2.9 (a)), qui peut ensuite être affinée à partir d'autres points de vue. Il peut ensuite modifier la position des autres membres du corps l'aide de lignes secondaires. Contrairement à *Differential Blending for Expressive Sketch-Based Posing* cette méthode est entièrement automatique, l'utilisateur n'a pas besoin de spécifier de correspondance entre la ligne d'action et une partie du corps du personnage, ni de pré-sélectionner manuellement les os qui seront bougés, puisqu'une seule ligne d'action va complètement modifier l'intégralité de la pose du modèle. De plus, cette méthode fonctionne même lorsque le dessin n'a pas des proportions identiques au modèle 3D en effectuant, à chaque étape, une reparamétrisation optimale de la correspondance entre la ligne d'action et les os du squelette. De manière à lever les ambiguïtés sur les significations 3D d'une même ligne 2D, sans avoir à faire des hypothèses sur la physiologie du personnage que l'on désire poser, la transformation que va subir le modèle par rapport à la ligne d'action doit se situer dans le plan de visualisation ; par conséquent l'utilisateur doit effectuer une rotation 3D à chaque fois qu'il désire effectuer une modification de la pose. De plus, en voulant automatiser au maximum le processus pour le simplifier, l'utilisateur se voit priver d'un contrôle précis du positionnement de chaque partie du corps.

*SketchiMo : Sketch-based Motion Editing for Articulated Characters* de Choi et al. [CiRL<sup>+</sup>16] utilise une idée assez proche de la ligne d'action, cependant au lieu d'essayer de représenter avec une ligne la position que doit prendre le personnage 3D cette méthode utilise des lignes pour représenter un déplacement que va devoir suivre ce personnage au cours du temps (fig. 2.9 (b)). L'utilisateur a la possibilité de définir des mouvements dans un espace global, affectant l'ensemble du modèle 3D, ou dans un espace local, en précisant manuellement la partie du corps qui devra suivre cette trajectoire. Un ensemble de différentes trajectoires peut être défini sur plusieurs membres du corps en même temps. Lorsque l'utilisateur définit un mouvement qui doit être suivi par une des extrémités du corps, le système calcule à l'aide de techniques de cinématique inverse comment modifier les paramètres du squelette pour obtenir ce mouvement. Malgré tout leur système souffre de quelques limitations : l'édition de l'orientation des parties du corps n'est pas possible avec 3 degrés de liberté actuellement, de plus, bien que cette méthode soit conçue pour l'animation, elle ne gère pas de nombreuses manipulations temporelles pourtant importantes dans ce domaine, telles qu'un ralentissement ou une accélération.

L'ensemble des systèmes présentés dans cette section permettent à l'utilisateur d'effectuer rapidement et simplement le repositionnement d'un modèle 3D par le biais de croquis 2D. Cependant, tout comme les méthodes dédiées à la reconstruction de bas-reliefs, elles n'effectuent pas une reconstruction complète d'un modèle 3D.

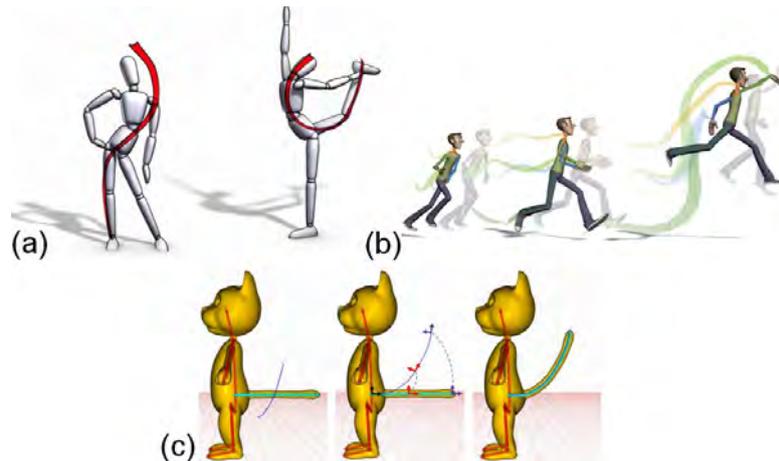


FIGURE 2.9 – Résultats produits par les méthodes qui utilisent des lignes d’action pour effectuer le repositionnement de modèles 3D : (a) *The Line of Action : an Intuitive Interface for Expressive Character Posing* [GCR13] permet de poser un personnage en une seule ligne, la représentation d’un déplacement que doit suivre un caractère est effectuée avec *SketchiMo : Sketch-based Motion Editing for Articulated Characters* [CiRL<sup>+</sup>16] par le biais de courbes (b) et *Differential Blending for Expressive Sketch-Based Po-sing* [OBP<sup>+</sup>13] utilisé pour repositionner un os préalablement sélectionné par l’utilisateur en un seul tracé(c).

## 2.4 Reconstruction à partir de croquis sous de multiples vues

Une façon d’obtenir les informations nécessaires à la reconstruction 3D est de ne pas se limiter à un seul croquis en entrée. En effet, si la reconstruction est complexe à partir d’un croquis, c’est parce que celui-ci ne représente pas pleinement les parties cachées de l’objet et ne donne aucune information de profondeur ; mais si l’on fournissait aux programmes ces informations, alors le problème de la reconstruction géométrique serait plus simple à résoudre. En utilisant plusieurs dessins de l’objet à reconstruire, sous différents angles de vue, on procure au système les données nécessaires. La multiplicité des angles de vue permet que toutes les parties de l’objet soient représentées sur au moins un des croquis, ainsi si un élément est caché sur un croquis, l’utilisateur fournira une esquisse d’un autre point de vue où il sera visible. *3D modeling with silhouettes* de Rivers et al. [RDI10] propose un moyen pour utiliser de multiples silhouettes comme outils de modélisation pour des objets manufacturés. Le modèle final est réalisé par assemblage d’éléments qui sont spécifiés par deux ou trois silhouettes dessinées vues de face, de profil et de haut, comme montré sur la fig. 2.10. L’utilisateur est donc invité à dessiner chacun des éléments les uns après les autres, selon ces trois points de vue spécifiques uniquement. Un modèle 3D est généré dès qu’une nouvelle silhouette est dessinée et

est créée comme étant l'intersection des 2 ou 3 silhouettes de cet élément projeté selon l'axe correspondant à leurs angles de vue. En restreignant les points de vue possibles pour la représentation des éléments, le système obtient les informations nécessaires pour mettre en correspondance ces silhouettes et calculer leur représentation 3D, cependant cela limite également la complexité que peut avoir chaque élément. En effet, il est alors impossible pour une partie du modèle 3D d'avoir des facettes non perpendiculaires à au moins un de ces 3 points de vue.

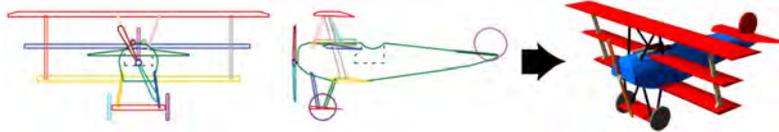


FIGURE 2.10 – Le modèle 3D est généré automatiquement par *3D modeling with silhouettes* [RDI10] en calculant l'intersection des silhouettes fournies par l'utilisateur.

Wang et al. proposent avec *Progressive sketching with instant previewing* [WZS19] un modeleur permettant à l'utilisateur de produire et de raffiner une forme 3D en fournissant des coupes transversales successives de la forme voulue sur trois plans orthogonaux (fig. 2.11). L'utilisateur commence par dessiner un contour fermé de la forme désirée sur un premier plan, et le système lui donne un aperçu de la forme reconstruite. Il peut ensuite modifier ce résultat en esquissant des coupes transversales sur les 2 autres plans ; plans dont il peut modifier la position par le biais d'une translation s'il désire définir de multiples coupes selon un même axe. Une série d'outils de sculpture 3D, tels que l'extrusion et le lissage, lui sont également fournis afin de pouvoir raffiner la forme reconstruite ; toutes ces opérations étant réalisées par le biais de courbes tracées directement sur la surface et interprétées par le système comme des commandes. Cependant, tout comme pour *3D modeling with silhouettes*, l'objet reconstruit doit être exprimable selon les 3 plans orthogonaux, qui correspondent à la vue de face, de côté et de haut, et ne permet pas l'utilisation de plans arbitraires.

*ShapeShop* de Schmidt et al. [SWSJ07] et *FiberMesh* de Nealen et al. [NISHA07] sont tous les deux partis d'un même principe. Plutôt que d'imposer à l'utilisateur les angles de vues sur lequel dessiner, laissons-le esquisser de manière incrémentale selon n'importe quel point de vue par le biais d'outils de rotation 3D. Dans un premier temps, l'utilisateur dessine un premier contour fermé à partir duquel un maillage 3D est obtenu par gonflement. L'utilisateur peut, par la suite, éditer ce maillage 3D et rajouter de nouvelles parties en dessinant de nouveaux contours. *ShapeShop* offre divers outils permettant à l'artiste de spécifier les opérations qu'il désire réaliser (extrusion, fusion, etc.) avec ses prochaines courbes. Un ensemble d'outils de manipulation 3D permettent également la modification du maillage, de son épaisseur, sa rotation, sa position. L'utilisateur peut donc reconstruire une large variété de formes en dessinant élément par élément, et selon

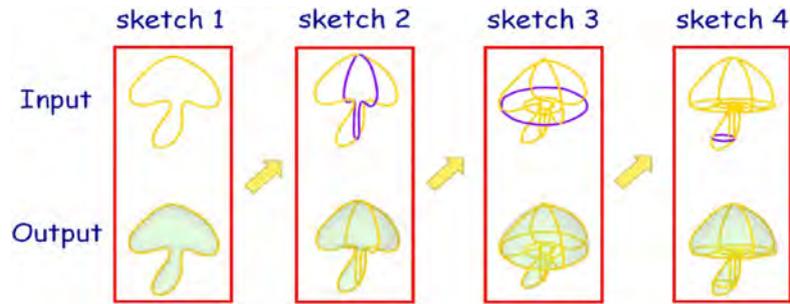


FIGURE 2.11 – Processus de modélisation avec *Progressive sketching with instant previewing* [WZS19] en dessinant successivement des coupes transversales sur les trois plans orthogonaux.

un angle de vue adaptée pour chacun. Il peut ensuite modifier manuellement les propriétés de l'élément créé pour qu'il corresponde à ses attentes. Dans *FiberMesh*, une fois que la première forme a été produite à partir du premier contour 3D, l'utilisateur a la possibilité de redessiner sur celle-ci des courbes de différents types, et sous différents points de vue, qui sont ensuite utilisées comme poignées de contrôle pour déformer en douceur la forme initiale, via une optimisation fonctionnelle, ou pour appliquer des opérations de modélisation 3D classique telle que l'extrusion ou la section du modèle 3D. Il a également la possibilité d'effacer toute courbe qu'il trace, intégralement ou en partie, de manière à limiter la modification apportée par celle-ci. Des exemples de résultats produits avec ces deux méthodes sont présentés fig. 2.12.

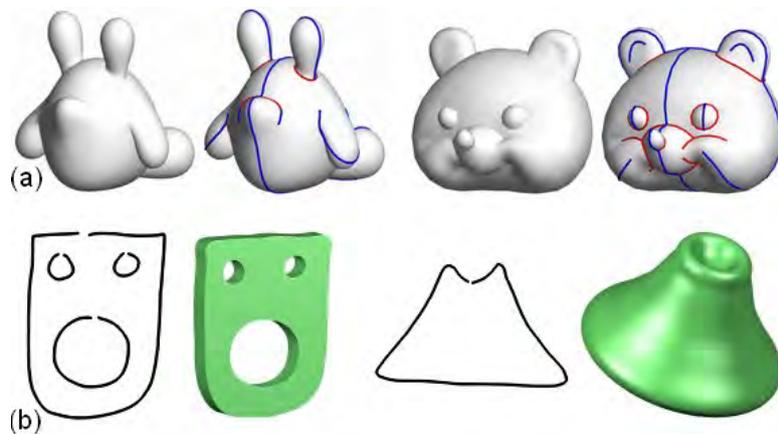


FIGURE 2.12 – Résultats produits avec *Fibermesh* [NISHA07] sur la première ligne (a), et formes simples créées avec *ShapeShop* [SWSJ07] seconde ligne (b).

Bien que ces deux méthodes offrent un contrôle énorme sur la forme obtenue, et per-

mettent la reconstruction d'une large variété de modèles, construire des formes extrêmement complexes, présentant un grand nombre d'entrelacements ou des formes englobant d'autres éléments par exemple, reste extrêmement difficile. De plus, l'ensemble des outils et des opérations 3D effectuées par l'utilisateur dans ces systèmes les rapproche des logiciels de modélisations classiques, et perd un peu de l'intérêt de la modélisation par croquis. Par ailleurs, l'utilisation de plusieurs dessins comme données d'entrée restreint l'accessibilité de ces systèmes aux utilisateurs capables de dessiner un même objet sous plusieurs points de vues différentes en s'assurant que chacune de ces représentations soit en accord avec les autres (dessiner chaque élément aux bonnes positions et avec les bonnes proportions selon chaque angle de vue).

## 2.5 Reconstruction de surface à partir d'un réseau de courbes 3D

Plutôt que de tenter de reconstruire directement une surface 3D à partir du dessin, certains chercheurs se sont orientés sur la reconstruction d'un réseau de courbes 3D. La création d'un modèle 3D basée sur une esquisse se trouve alors divisée en deux étapes : la détermination d'un réseau de courbes 3D à partir d'un croquis, suivi d'une construction de surface à partir du réseau de courbes 3D précédemment déterminé. Ce choix est motivé par plusieurs raisons. Se focaliser uniquement sur la reconstruction de courbes permet l'utilisation de types de dessin plus explicite contenant un plus grand nombre de courbes pour représenter l'objet. C'est le cas principalement pour les dessins effectués lors de la conception d'objets manufacturés, dont la géométrie est décrite en traçant davantage de courbes afin de fournir des informations sur le parallélisme ou l'orthogonalité des parties de l'objet par exemple. De plus, la reconstruction de surface à partir d'un réseau de courbes 3D est un domaine de recherches proposant déjà des solutions, telles que [PLS<sup>+</sup>15] et [ZZCJ13], permettant ainsi aux chercheurs de se focaliser uniquement sur le problème de détermination de courbes 3D à partir d'un dessin 2D.

Une première approche pour ce problème est de proposer un moyen simple de dessiner des courbes directement placées dans l'espace 3D. C'est ce que permettent de faire *ILoveSketch* de Bae et al. [BBS08] et *Analytic Drawing of 3D Scaffolds* de Schmidt et al. [SKSK09].

*ILoveSketch*, propose une interface de dessin avec une gestuelle permettant de dessiner les courbes directement dans l'espace 3D. Conçue pour être utilisée avec une tablette, elle permet à l'utilisateur d'effectuer avec son stylet l'ensemble des manipulations nécessaires au dessin 3D (rotation 3D, sélection du plan dans lequel dessiner, etc.) par le biais d'une gestuelle composée uniquement de 5 actions simples. Le positionnement en 3D des courbes dessinées se fait à l'aide de méthodes basées sur la géométrie épipolaire en prenant en compte l'angle de vue actuel au moment du tracé de la courbe 2D.

*Analytic Drawing of 3D Scaffolds* se base sur les principes du *dessin analytique* qui est un processus de dessin permettant de représenter une forme 3D. L'utilisateur commence par tracer un échafaudage de lignes de régulation qui expriment les relations 3D

dans le dessin. Ces lignes servent alors de contraintes permettant d’esquisser de manière très précise les courbes composant l’objet. Lorsque l’artiste trace dans l’interface une première courbe, le système calcule automatiquement les principaux points de fuite qui sont ensuite utilisés pour créer des lignes directrices en esquissant un trait allant jusqu’à un de ces points. L’utilisateur réalise donc avec ces lignes directrices un support sur lequel il peut par la suite tracer des segments 3D sans ambiguïté de profondeur ou d’angles d’intersections.

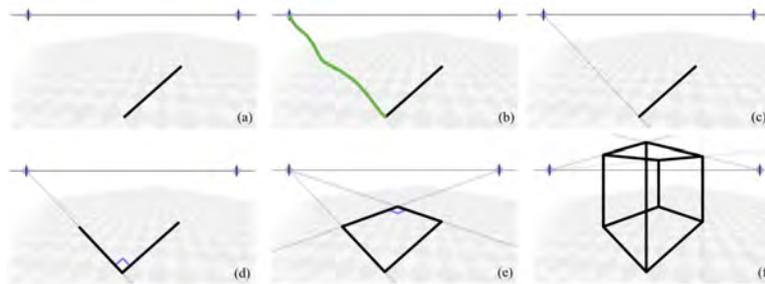


FIGURE 2.13 – Création d’un échafaudage permettant à l’utilisateur de dessiner un cube en 3D avec la méthode *Analytic Drawing of 3D Scaffolds*[SKSK09].

Cette approche permet aux croquis de l’utilisateur d’être directement interprétés en 3D, cependant elle nécessite que l’artiste effectue des actions avant de tracer les courbes, de manière à les placer correctement. Bien que ces actions soient relativement simples (une rotation 3D du dessin pour changer l’angle de vue, l’ajout de nouvelles lignes directrices, etc.), elles rajoutent des étapes au processus de création déjà complexe, et perdent donc la simplicité qu’un simple croquis a et qui est à l’origine de ce domaine de recherche.

La seconde approche consiste à déduire le positionnement des courbes 3D à partir d’un unique dessin intégralement réalisé dans le domaine 2D. Bien que *BendFields* de Larussi et al. [IBB15] ne permette pas réellement la reconstruction de courbes 3D, elle permet un rendu 3D à partir d’un croquis selon l’angle de vue du dessin, un peu à l’image des bas-reliefs. Il s’agit d’un système semi-interactif où l’utilisateur commence par charger le dessin de conception et effectuer une distinction manuelle entre les courbes qui correspondent aux contours de l’objet, et celles qui servent à indiquer sa courbure. Le programme extrapole à partir de cette unique information la courbure 3D et la normale à chaque pixel. L’utilisateur est ensuite invité à lever manuellement les ambiguïtés entre les zones concaves et convexes. Ces informations permettent alors au système d’appliquer dynamiquement un ombrage sur un dessin, permettant ainsi un rendu proche d’un modèle 3D.

*True2Form* de Xu et al. [XCS<sup>+</sup>14] cherche à détecter automatiquement les mêmes indices de régularités géométriques dans les dessins de conception que celle que l’Homme emploie pour générer une image 3D mentale de l’objet esquissé (fig. 2.14). Ces indices

suivent un ensemble de règles qui permettent de déduire automatiquement l'orthogonalité, le parallélisme, la symétrie et la planarité que les courbes doivent avoir en 3D. Bien que leur méthode cherche l'automatisme à partir d'une vue unique, l'utilisateur peut changer occasionnellement d'angle de vue si nécessaire. Il lui est également demandé de lever manuellement les ambiguïtés qu'il peut y avoir. Il doit, par exemple, distinguer les croisements 2D qui correspondent à des intersections 3D de ceux provoqués par des occultations, et peut changer les zones concaves en convexes, ou inversement, si le choix effectué par l'algorithme pour la reconstruction 3D ne convient pas.

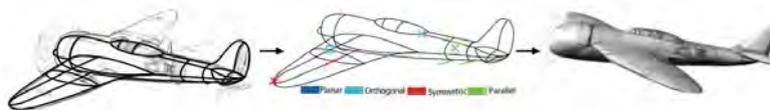


FIGURE 2.14 – À partir du dessin en entrée, *True2Form* [XCS<sup>+</sup>14] détecte automatiquement les indices de régularité géométrique dans le dessin de manière à produire le modèle 3D.

Ces deux approches permettent la reconstruction de courbes en 3D, que ce soit par le biais d'une interface de dessin 3D ou par reconstruction à partir d'un croquis 2D. Cependant, l'idée d'utiliser ces courbes amène malgré tout une limitation sur les objets reconstituables. En effet, ce genre de représentation est dédié aux objets manufacturés, dont la courbure est facilement représentable par quelques courbes. Ces méthodes ne permettent donc pas la reconstruction d'objets organiques plus complexes. De plus, tout le monde ne connaît pas spécifiquement le style de dessin des concepteurs qui est nécessaire pour ces méthodes requérant des croquis suffisamment descriptifs pour produire un bon résultat 3D.

## 2.6 Reconstruction guidée par des données

La reconstruction d'un maillage 3D à partir d'un seul dessin est complexe car la quantité d'information fournie par un dessin unique est très faible par rapport à celle apportée par l'objet 3D. Pour résoudre ce problème, il est nécessaire d'obtenir les informations d'une façon ou d'une autre. Contrairement aux méthodes qui se servent de dessins sous de multiples vues pour combler ce manque, une autre solution consiste à être guidée par des données.

### 2.6.1 Connaissances a priori

Le premier type de données sur lequel on peut s'appuyer pour la reconstruction consiste à avoir des a priori, par exemple savoir quel sera le type d'objet dessiné ou supposer que celui-ci présentera une géométrie particulière.

Un a priori que l'on retrouve dans plusieurs travaux est la présence d'une symétrie entre plusieurs composants du modèle 3D, ou même l'auto symétrie d'un seul composant. *SymmSketch* de Miao et al. [MHZ<sup>+</sup>15] et *Modeling 3D animals from a side-view sketch* de Entem et al. [EBC<sup>+</sup>15] utilisent tous les deux ce principe.

*SymmSketch* utilise un plan de symétrie prédéfini sur lequel l'utilisateur doit dessiner les courbes des éléments symétriques afin de générer un maillage triangulaire. L'information de profondeur est ensuite calculée en utilisant les propriétés de symétrie miroir. Afin de permettre une plus grande liberté à l'utilisateur pour contrôler les caractéristiques géométriques locales, le système crée les formes à partir de quatre lignes de construction ; ainsi pour chaque élément composant la forme finale, une surface paramétrique est créée en utilisant une paire de lignes symétriques et une paire de courbes génériques. Cependant, leur méthode requiert que l'utilisateur dessine les courbes dans un ordre spécifique pour que le calcul de la forme se fasse correctement. De plus, un seul plan de symétrie est supporté.

*Modeling 3D animals from a side-view sketch* est une méthode qui se base sur deux a priori complémentaires : la forme esquissée est celle d'un animal vu de côté, et elle présente des éléments symétriques entre eux. Ces a priori sont complémentaires, car la quasi-globalité des animaux possède des parties symétriques entre elles, par exemple leurs pattes ou leurs oreilles. De plus, en forçant l'angle de vue du dessin à être vue de côté, on impose que le plan dans lequel est réalisée l'esquisse soit le plan de symétrie de l'animal. Ce système va séparer le croquis en éléments distincts, détecter la partie principale de l'animal, le torse, située dans le plan sagittal et s'en servir pour identifier les paires d'éléments symétriques et définir un ordre de profondeurs entre eux. Pour chacun de ces éléments, une surface 3D est reconstruite en calculant le squelette à l'aide de leur axe médian. Le modèle final est ensuite obtenu en assemblant ces surfaces. Des résultats obtenus avec ce système sont présentés fig. 2.15. Pour rendre la reconstruction possible, il est malgré tout nécessaire que les éléments composant la forme finale possèdent un axe médian planaire, situé dans le plan vertical. Il ne faut pas non plus qu'un des éléments contienne des auto-occultations.

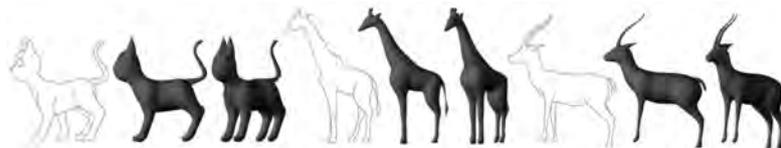


FIGURE 2.15 – Exemple de modèles produits par *Modeling 3D animals from a side-view sketch* [EBC<sup>+</sup>15].

L'utilisation de la symétrie se trouve également dans *RigMesh* de Borosán [BJD<sup>+</sup>12]. Ce système permet la création de modèles articulés, c'est-à-dire avec un squelette, prêt pour l'animation. Le dessin fourni par l'utilisateur est alors décomposé en régions qui peuvent être modélisées par un cylindre généralisé. Lorsqu'une partie du dessin présente une symétrie locale, elle est alors identifiée comme une partie cylindrique, et son squelette

est calculé. Les parties sans symétrie sont, quant à elles, identifiées comme étant une zone d'attachement entre les parties cylindriques. Le système propose également des opérations de coupe et de fusion d'éléments qui permettent la création de nouveaux modèles à partir de modèles déjà existants. La pose du maillage reconstruit peut ensuite être modifiée par l'utilisateur. Cette méthode va cependant échouer à reconstruire une forme désirable lorsque les éléments du dessin ne contiennent pas d'axe de symétrie évident.

Le second type d'a priori sur lequel des méthodes peuvent se baser est la nature de l'objet dessiné. En se limitant à la reconstruction d'un certain type de forme, ces méthodes sont capables de produire des résultats qui correspondent aux attentes de l'artiste. Cette limitation peut être assez générique, tel que se restreindre à des objets lisses et arrondis, tout comme elle peut être très spécifique, comme se limiter aux fleurs ou aux corps humains.

*SmoothSketch* de Karpenko et Hughes [KH06] et *Free-Form Sketching of Self-Occluding Objects* de Cordier et Seo [CSPyN11] sont toutes les deux des méthodes qui permettent l'inflation de surfaces lisses à partir des contours présents sur le croquis. *SmoothSketch* permet la reconstruction de formes complexes avec plusieurs jonctions (jonctions en T ainsi que jonctions cuspidées). Pour chaque contour fermé, un maillage est généré par inflation. Ceux-ci sont ensuite assemblés pour former le modèle final. *Free-Form Sketching of Self-Occluding Objects* est dédié à la reconstruction d'objets présentant des auto-occultations. Dans un premier temps, le système effectue la complétion du croquis, en calculant les contours cachés de l'élément, puis il détermine un ordre de profondeur entre les courbes du contour. De cette manière, il peut ensuite effectuer une reconstruction 3D de l'objet en plaçant les courbes de manière à respecter l'ordre de profondeur défini précédemment et éviter les auto-intersections de l'élément 3D. Ces deux méthodes souffrent malgré tout d'un manque de contrôle réel sur la forme reconstruite et notamment sur son volume ou son placement en profondeur, pouvant ainsi donner l'impression que les résultats obtenus sont légèrement aplatis, comme on peut le voir fig. 2.16. De plus, faire des formes extrêmement complexes avec ces méthodes, tels que des objets avec beaucoup d'entrelacements, est impossible.

Limité à une catégorie d'objets plus spécifique, *Floral diagrams and inflorescences* de Ijiri et al. [IOOI05] propose un système interactif permettant à l'utilisateur de construire des modèles 3D de fleurs de qualité. Dans ce système l'utilisateur participe à toutes les étapes du processus de création. Il commence par dessiner le diagramme floral, une représentation schématique de l'organisation des pièces florales d'une fleur, puis créer chacun des composants 3D de la fleur. Ensuite, il dessine l'inflorescence de la fleur utilisée par la suite pour placer l'ensemble des éléments et constituer le modèle 3D final (fig. 2.17). Cependant, comme ce système nécessite un grand nombre d'interactions avec l'utilisateur, la reconstruction d'une seule fleur requiert énormément de temps (30 min ou plus de manière générale).

*3D layered articulated object from a single 2D drawing* de Noé [Noé17] est spécialisé dans la reconstruction d'objets articulés constitués de couches rigides. L'utilisateur

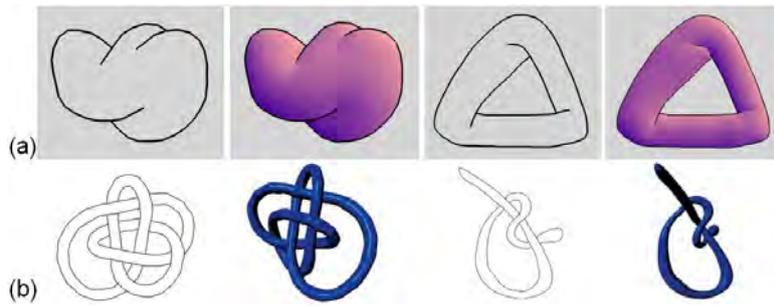


FIGURE 2.16 – Résultats obtenus par les méthodes *SmoothSketch* [KH06] (a) et *Free-Form Sketching of Self-Occluding Objects* [CSPyN11] (b). Bien que les reconstructions soient conformes au dessin, elles manquent encore un peu de profondeur et donnent une impression d’aplatis.

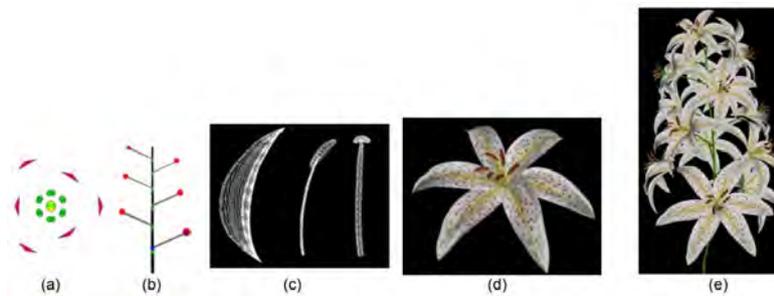


FIGURE 2.17 – Avec le système *Floral diagrams and inflorescences*[IOOI05] l’utilisateur fournit les informations sur la structure de la fleur à partir d’un diagramme floral (a) et en dessinant l’inflorescence qu’elle doit avoir(b). Il crée ensuite les modèles géométriques (c) utilisés pour constituer la fleur (d) qui sont alors placés pour former le modèle final (e).

indique sur l’image d’entrée les courbes correspondant aux éléments circulaires et non circulaires de chaque couche, et le système est ensuite capable de reconstruire et positionner correctement chacune d’entre elles en 3D. Cependant, l’objet représenté sur l’image d’entrée doit être nécessairement vu de trois quarts, et la visibilité de chaque partie articulée doit être telle que la fin de chaque élément est représentée par une courbe dont la courbure est éloignée de zéro. En effet, la courbe doit pouvoir être approximée par une courbe circulaire, ce qui rend cette méthode extrêmement limitée.

En se servant de l’a priori que le dessin représente un personnage, et que celui-ci est forcément dessiné vue de face, *Automatic Single-View Character Model Reconstruction* de Buchanan et al. [BMD13] permet la reconstruction complètement automatique d’un maillage triangulaire 3D représentant le personnage esquissé. Cette méthode commence par générer un squelette à partir du croquis, puis, en approximant l’orientation 3D et en

analysant les propriétés de courbure des lignes, définit des points autour desquels créer des coupes transversales et construire le maillage final. Le modèle résultant est ensuite texturé en utilisant les couleurs de l'image d'entrée. Cependant, cette méthode a du mal à reconstruire des éléments fins, tel que des ailes. Il est également nécessaire que le dessin ne contienne qu'un nombre minimal d'auto-occultations, et que les membres du personnage ne se touchent pas.

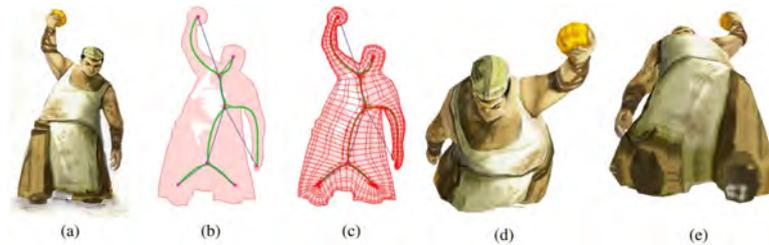


FIGURE 2.18 – Processus de création du maillage d'un personnage avec , *Automatic Single-View Character Model Reconstruction* de Buchanan et al. [BMD13]. À partir du dessin en entrée (a) le système calcule un squelette (b) qui est utilisé pour produire le modèle final (c),(d),(e).

Il ne s'agit pas de la seule méthode pleinement automatique à l'aide d'a priori. *Automatic structuring of organic shapes from a single drawing* de Entem et al. [EPB<sup>+</sup>19] permet l'extraction automatique des couches qui composent la forme représentée en se limitant à la reconstruction des formes organiques. Par forme organique, on entend : toute forme 3D dont les éléments, ainsi que les connexions, sont lisses. Cette méthode se sert de l'axe médian afin d'effectuer la décomposition et la complétion du dessin en régions fermées, ainsi que pour calculer les informations sur la structure et la position des pièces composant le modèle. Une méthode d'inflation est ensuite utilisée pour obtenir un élément 3D à partir de chacune des régions obtenues précédemment. Ces dernières sont ensuite placées en 3D conformément à la structure déterminée. Des résultats produits avec cette méthode sont montrés fig. 2.19. Cette méthode ne permet cependant pas la reconstruction d'objets dont les parties sont disposées de manière cyclique les unes sur les autres. De plus, l'étape de décomposition peut échouer si le dessin présente des ambiguïtés, par exemple lorsque l'angle de vue choisi par l'artiste pour représenter l'objet ne permet pas de bien détecter et définir les jonctions en T.

Certaines méthodes ont poussé plus loin l'a priori sur la nature de l'objet représenté, en se servant d'une base de données 3D composée de modèles 3D de cette nature pour aider à la reconstruction.

*MotionMaster* de Li et al. [LGY<sup>+</sup>06] est un système qui permet d'animer un mannequin en lui faisant effectuer une chorégraphie de Kung-Fu à partir de croquis. L'utilisateur commence par dessiner un mannequin en pose T en esquissant bien les joints entre chaque membre. Il fournit ensuite plusieurs croquis représentant les différentes postures,

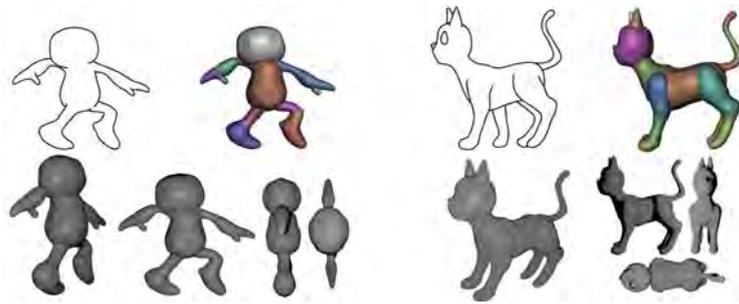


FIGURE 2.19 – Deux exemples de modèles organiques reconstruits automatiquement à partir d’une seule image avec la méthode *Automatic structuring of organic shapes from a single drawing*. [EPB<sup>+</sup>19].

avec les joints toujours représentés, qui consistent en la position de départ et de fin du mouvement que doit effectuer le mannequin. Celui-ci est par des lignes représentant la trajectoire que les articulations doivent suivre. Le système va alors chercher dans la base de données un mouvement dont la position et la trajectoire sont similaires à celui esquissé, et va le raffiner de manière à produire l’animation du mannequin correspondant au croquis.

Toujours dans le thème de la reconstruction d’êtres humains, *Efficient Sketch-Based Creation of Detailed Character Models through Data-Driven Mesh Déformations* de Kazmi et al. [KYY<sup>+</sup>15] fait usage d’une base de données de personnages humains 3D pour reconstruire un corps 3D extrêmement détaillé (fig. 2.20). En effet, l’Homme représenté sur le croquis n’est pas uniquement décrit par le biais des courbes de sa silhouette, mais également par de nombreuses courbes servant de contours suggestifs et oclusifs, apportant les détails sur la forme à l’intérieur de la silhouette. Leur système va ensuite effectuer une recherche dans la base de données pour trouver le modèle 3D correspondant le plus à ce dessin en entrée. Une fois celui-ci trouvé, une correspondance sera effectuée entre la silhouette du modèle 3D et celle du dessin, de manière à pouvoir modifier la pose du modèle et d’appliquer une déformation de Laplacien au maillage pour qu’il corresponde à l’esquisse de l’utilisateur.

L’utilisation d’une base de données 3D pour guider la reconstruction est utilisable pour la reconstruction d’objets en tout genre. *Model-driven Sketch Reconstruction with Structure-oriented Retrieval* de Li et al. [LHZ<sup>+</sup>16] s’en sert pour trouver la structure entre les plans qui composent un objet manufacturé dessiné par l’utilisateur, tel qu’une chaise ou une table ; cette structure est ensuite utilisée pour effectuer une rétroprojection du dessin (fig. 2.21). Le système effectue une récupération de modèle dans la base de données de manière à trouver, dans la base, le modèle 3D le plus proche de celui esquissé, et le retourne dans l’orientation correspondant à celle de l’objet dessiné. Il se sert ensuite de ce modèle pour détecter les plans 3D qui le composent ainsi que la structure de l’objet esquissé. Une projection du dessin sur ces plans est ensuite effectuée, obtenant ainsi un pseudo-modèle 3D qui sera utilisé pour permettre la rétroprojection selon un autre angle

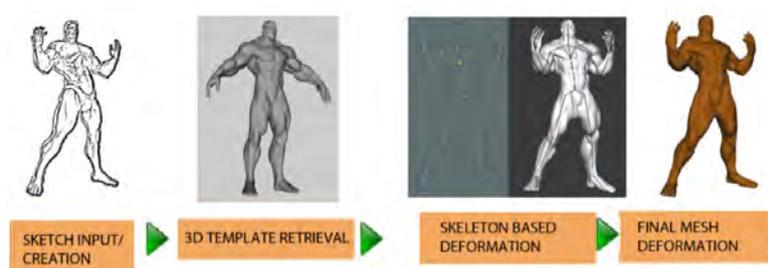


FIGURE 2.20 – Processus simplifié de la méthode *Efficient Sketch-Based Creation of Detailed Character Models through Data-Driven Mesh Déformations* [KYY<sup>+</sup>15]. Un modèle référent est recherché dans la base de données à partir du dessin, et celui-ci est ensuite déformé pour correspondre au croquis.

de vue.



FIGURE 2.21 – Exemples de rétroprojection sous d'autres angles de vues obtenues à l'aide de *Model-driven Sketch Reconstruction with Structure-oriented Retrieval* [LHZ<sup>+</sup>16].

Comme le montrent ces travaux, se baser sur des a priori permet de rendre soluble le problème de reconstruction 2D à partir d'une image. Cependant, il limite également grandement ces méthodes, qui se retrouvent dans l'incapacité de reconstruire des formes qui ne répondent pas à ces a priori. Cette limitation est encore plus vraie pour les méthodes se servant d'une base de données d'objets 3D pour guider la reconstruction. En effet, ces systèmes requièrent que l'objet dessiné corresponde non seulement à la catégorie pour laquelle ils sont conçus, mais il faut également qu'il soit proche d'un des modèles 3D présents dans la base de données. Dans le cas contraire, la reconstruction ne serait pas de qualité suffisante, voir pourrait complètement échouer.

## 2.6.2 Composition de modèles préexistants

Malgré tout, se servir d'un ensemble de modèles 3D préexistants pour guider la reconstruction est une idée qui a séduit un grand nombre de chercheurs. Certains ont même poussé le principe encore plus loin. La reconstruction n'est plus simplement guidée par un modèle 3D, mais utilise de multiples modèles 3D préexistants pour composer le résultat final lui-même.

*Geosemantic Snapping for Sketch-Based Modeling* de Shtof et al. [SAG<sup>+</sup>13] effectue la composition du modèle final à partir de primitives 3D simples. Après avoir fourni le dessin qu'il désire reconstruire, l'utilisateur sélectionne une des 6 primitives géométriques disponibles dans le système et la fait glisser sur le contour de l'élément du dessin correspondant. Le système déforme alors la primitive de manière à la faire correspondre au dessin. Les éléments 3D sont automatiquement orientés et attachés les uns aux autres à chaque ajout d'une nouvelle primitive par l'utilisateur (fig. 2.22). Cette méthode est cependant limitée aux pièces mécaniques qui sont des objets pouvant être décomposés en géométrie simple (cube, sphère, cône, etc.).

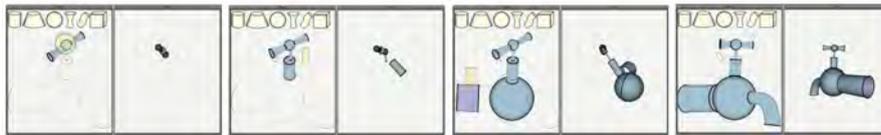


FIGURE 2.22 – Avec *Geosemantic Snapping for Sketch-Based Modeling* [SAG<sup>+</sup>13] l'utilisateur sélectionne et dispose des primitives géométriques sur le croquis. Celles-ci sont placées de manière à correspondre aux contours dessinés et à s'assembler ensemble.

Pour permettre la production de résultats composés d'éléments plus complexes que de simples primitives géométriques, une solution adoptée par de nombreux travaux est d'élargir la base de données de composants 3D. *Interactive sketching of urban procedural models* de Nishida et al. [NGDA<sup>+</sup>16] permet, en combinant un ensemble de grammaire procédurale, la reconstruction d'un bâtiment 3D (fig. 2.23). L'utilisateur dessine chaque composant du bâtiment dans un ordre précis. Pour chacun d'entre eux, le système retourne la grammaire procédurale permettant de produire le résultat le plus proche de celui esquissé par l'utilisateur. Ces grammaires sont ensuite fusionnées en une seule représentant le bâtiment dans son intégrité.

*Sketch-to-Design* de Xie et al. [XXM<sup>+</sup>13] se sert des informations sur le contexte pour composer le modèle final (fig. 2.24 (a)). L'utilisateur commence par sélectionner un modèle 3D référent, tel qu'une chaise, qui servira de base à sa composition. Il dessine ensuite, selon l'angle de vue de son choix, sur le composant du modèle actuel qu'il désire changer. Le système, connaissant la nature du modèle initiale, est capable de comprendre à quel composant de ce modèle correspond le dessin de l'utilisateur. Il cherche alors dans sa base de données 3D un modèle possédant un composant se rapprochant de

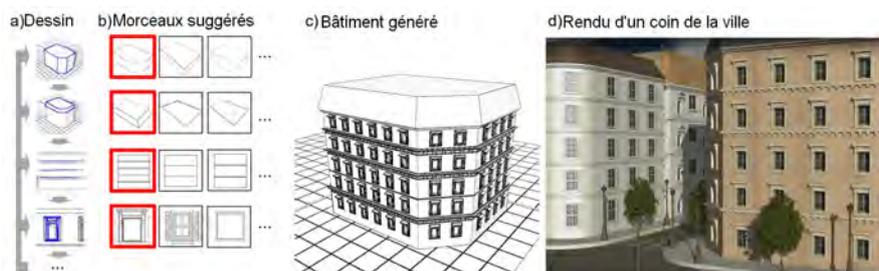


FIGURE 2.23 – Processus de génération de *Interactive sketching of urban procedural models* [NGDA<sup>+</sup>16]. À partir du dessin des éléments du bâtiment fournis par l'utilisateur (a) le système recherche, dans sa base, les morceaux les plus proches (b) et s'en sert pour générer le bâtiment (c).

celui esquissé par l'artiste. Le modèle initial est ensuite modifié en remplaçant la partie d'origine par celle retrouvée dans la base. Similairement *Modeling By Example* de Funkhouser et al. [FKS<sup>+</sup>04] et *Sketch-Based Search and Composition of 3D Models* de Lee et Funkhouser [LF08] permettent la composition de modèles complexes et variés (fig. 2.24 (b) et (c)). Le premier système possède une base de données de modèles complets, dans laquelle l'utilisateur peut chercher les éléments de son choix à l'aide d'un dessin ou d'un champ de texte. Il dispose ensuite d'outils permettant de couper du modèle retrouvé la partie 3D qu'il désire incorporer au modèle original, à l'endroit spécifié manuellement par l'utilisateur. Le second système, lui, rend la composition plus automatique : la base de données est constituée de parties prédécoupées, et non pas de modèles complets, supprimant ainsi toutes les étapes de découpe manuelle à faire par l'utilisateur. Le positionnement sur le modèle des parties récupérées dans la base est fait automatiquement à l'endroit où l'utilisateur a dessiné l'élément. De plus, cette seconde méthode ne se limite pas à la reconstruction d'un modèle 3D composé de multiples éléments, mais permet également de réaliser des scènes complètes où chaque composant correspond en réalité à un objet de la scène tel qu'un bureau, une chaise et un écran dans une salle.

La reconstruction d'une scène par composition, plutôt que par la modélisation d'un seul objet 3D, est d'ailleurs un sujet qui a été à l'origine de plusieurs travaux. Avec *MagicCanvas* de Shin et Igarashi [SI07] l'utilisateur produit une scène en dessinant les objets qui la composent les uns après les autres (fig. 2.25 (a)). Ce système utilise une base de données où chaque objet 3D est associé à 16 projections 2D correspondant à 16 points de vue différents. Chaque fois qu'un objet est dessiné, il recherche dans la base de données la projection 2D la plus proche de ce dessin et en déduit l'objet 3D associé. Ce modèle est alors placé dans la scène à l'endroit où le dessin a été effectué. À chaque récupération d'un modèle depuis la base de données, un ensemble d'options est proposé à l'utilisateur dans deux panneaux différents. Le premier permet de changer l'objet récupéré, permettant à l'utilisateur de sélectionner un autre modèle parmi l'ensemble



FIGURE 2.24 – Exemples de modèles composés à l’aide de différentes méthodes : (a) une chaise réalisée avec *Sketch-to-Design* [XXM<sup>+</sup>13] , (b) une chaise conçue avec *Modeling By Example* [FKS<sup>+</sup>04] et (c) un pégase ainsi qu’une salle bureautique créés avec *Sketch-Based Search and Composition of 3D Models* [LF08].

des candidats récupérés dans la base, ce qui est utile lorsque le résultat par défaut ne correspond pas aux attentes de l’artiste. Le second permet de modifier l’orientation de l’objet dans la scène.

*Sketch2Scene* de Xu et al. [XCF<sup>+</sup>13] propose une solution plus automatique. Le système a appris les relations sémantiques entre les éléments pouvant composer une scène en se servant d’un ensemble de scènes 3D comme données d’apprentissage. Pour récupérer un modèle 3D depuis la base de données, le système se servira alors à la fois de l’esquisse 2D de cet élément et des autres composants 3D déjà récupérés et de leurs esquisses. En ayant plusieurs éléments comme données d’entrée, il est capable de déterminer la nature du nouvel objet en se servant des relations sémantiques apprises pour trouver quel élément 3D correspond à la disposition de l’objet esquissée par l’utilisateur par rapport aux autres éléments déjà trouvés. Ainsi la sélection du bon modèle 3D et son positionnement sont ici complètement automatiques. Des exemples de scènes reconstruites avec cette méthode sont présentés fig. 2.25 (b).

L’ensemble de ces méthodes souffre cependant d’une limitation proche de celles se servant d’a priori pour effectuer la reconstruction 3D. Pour composer le modèle final, ces systèmes utilisent des modèles 3D préexistants ; cependant cela signifie également que tous les modèles dont un ou plusieurs des composants ne correspondent pas à ceux présents dans la base ne sont pas reconstructibles. De plus, la création de la base de données 3D sur laquelle reposent ces méthodes requiert souvent beaucoup de temps et d’effort.

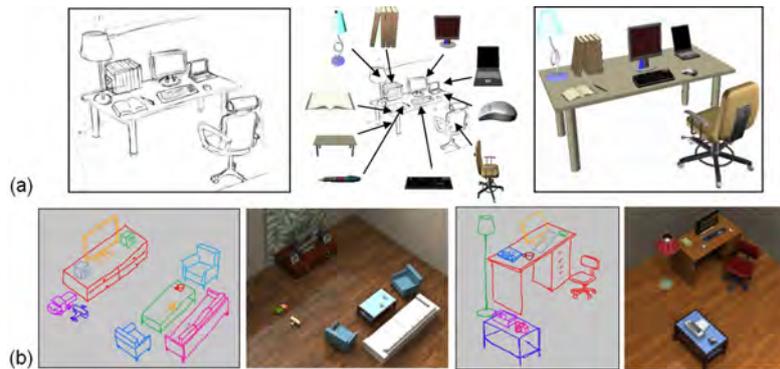


FIGURE 2.25 – La composition de scènes avec différentes méthodes : (a) salle bureautique conçue avec *MagicCanvas* [SI07], (b) 2 scènes modélisées avec *Sketch2Scene* [XCF<sup>+</sup>13].

### 2.6.3 Utilisation de modèles 3D comme support

Être guidé par des données 3D dans une base apporte un moyen d’obtenir des informations supplémentaires rendant le problème de reconstruction plus simple. Cependant cela nécessite la présence, dans la base de données du système, d’un modèle 3D correspondant à l’objet esquissé par l’artiste, ce qui limite l’utilisation de tels systèmes au contenu de leur base. De manière à pallier ce problème, certains systèmes ont décidé de ne pas dépendre d’une base de données 3D préalablement constituée, mais de demander à l’utilisateur lui-même de fournir en données d’entrée, aux côtés du dessin, les modèles 3D qui serviront à guider la reconstruction.

Avec *SecondSkin* de De Paoli et Singh [PS15] l’utilisateur peut créer un ensemble de structures en couches, en dessinant, des traits 2D sur un modèle 3D préalablement chargé. Leur méthode analyse les traits de l’esquisse et les classe parmi quatre types de courbes (normal à la géométrie 3D, tangent à la géométrie 3D, projection d’un élément, contour d’un élément) en se servant de la relation entre les traits 2D et la géométrie 3D sous-jacente. Une fois la classification effectuée, le système produit des courbes 3D qui servent à générer des patches de Coons et des volumes 3D correspondant aux différentes couches de la structure. Puisque cette méthode est destinée à la reconstruction d’objets à multiples couches, elle est parfaitement adaptée à la reconstruction d’armures ou d’autres tenues (fig. 2.26). Cependant la reconstruction n’est pas correcte lorsque les courbes dessinées par l’utilisateur ne sont pas classées correctement. Ceci arrive lorsqu’il y a de nombreuses occultations entre les pièces à l’intérieur de la géométrie 3D, ou lorsque l’angle de vue utilisé par l’utilisateur pour les esquisser est ambigu.

*Skippy* de Vojtěch et al. [KYC<sup>+</sup>17] permet à l’utilisateur de charger n’importe quelle scène 3D, composée d’un ou plusieurs modèles, afin de servir de support pour la reconstruction d’une ou de plusieurs courbes 3D (fig. 2.27). L’artiste commence par tracer une courbe en 2D à partir d’une seule vue du modèle. Cette courbe est ensuite convertie



FIGURE 2.26 – Modèles créés par des utilisateurs à l’aide de *SecondSkin* [PS15].

en courbes 3D passant par la scène actuelle. Un graphe de segments 3D est construit à partir de toutes les positions 3D possibles sur la scène de chaque groupe de points. Ce graphe est ensuite utilisé afin de sélectionner les meilleurs segments 3D pour former une courbe 3D complète. L’utilisateur dispose d’un moyen de contrôle sur le résultat puisqu’il peut manuellement choisir une autre solution du graphe comme étant la reconstruction qu’il désire. Si la scène 3D ne permet pas de positionner les courbes comme le souhaite l’artiste, ce dernier a la possibilité d’ajouter ce qu’ils ont appelé une *géométrie transitoire*, c’est-à-dire un objet géométrique 3D simple, tel qu’une sphère, pour ajouter plus de support pour la reconstruction. Bien que cette méthode soit limitée à la reconstruction de courbes 3D, elle permet malgré tout plusieurs applications, notamment elle simplifie la création d’un chemin 3D que doit suivre un objet dans la scène (une caméra par exemple).

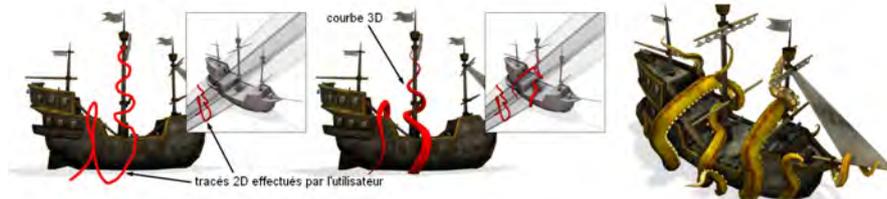


FIGURE 2.27 – *Skippy* [KYC<sup>+</sup>17] permet à l’utilisateur de créer facilement une courbe 3D en s’appuyant sur un modèle préalablement chargé.

*Model-guided 3d sketching* de Xu et al. [XFZ<sup>+</sup>19] utilise également un modèle 3D préalablement chargé pour déduire la position 3D des courbes dessinées par l’utilisateur. Cependant, contrairement aux deux méthodes précédentes, les courbes ne sont ici pas forcément placées sur le modèle 3D. En effet, cette méthode repose sur la détection automatique des plans 3D, sur lesquels les courbes dessinées doivent se positionner, en s’aidant des plans présents dans le modèle référent. Pour permettre leurs détections, il est demandé à l’utilisateur de dessiner les courbes plan par plan. Lorsque ce dernier esquisse les traits d’un nouveau plan, le système lui demande confirmation afin de s’assurer que

le plan 3D ainsi créé correspond à ces attentes. L’artiste peut également sélectionner un plan 3D déjà existant sur le modèle d’origine comme support sur lequel dessiner. De cette manière, cette interface permet de dessiner facilement sur et autour du modèle 3D chargé, et permet donc la modification ou la création d’extensions de ce dernier afin de modifier son design. Cette méthode se limite cependant au placement 3D du dessin de l’utilisateur, et n’effectue pas une reconstruction du modèle 3D. Elle est également limitée aux objets décomposables en plans 3D, et ne permet pas la création de formes arrondies.

Plutôt que d’utiliser des modèles 3D comme données d’entrée, des travaux se sont axés sur une reconstruction guidée par des données 3D brut (fig. 2.28). *Sketch-based guided modeling of 3D buildings from oriented photos* de Schwärzler et al. [SKMW17] permet la reconstruction 3D de bâtiments à l’aide de données photogrammétriques. Ces données consistent en un ensemble de photos du même bâtiment, prises sous différents angles de vues, et superposées à l’aide des techniques *Structure from Motion*, donnant les emplacements et les orientations à partir desquels les photos ont été prises et constituant un nuage de points 3D par projection des caractéristiques détectées dans les images dans l’espace 3D. Ce nuage de point 3D est ensuite utilisé pour définir le contexte dans lequel l’artiste va esquisser des polygones, permettant ainsi d’en déduire leurs positions 3D. L’utilisateur va donc dessiner l’ensemble des plans 3D constituant le bâtiment, qui seront ensuite texturés automatiquement en utilisant les informations présentes dans les photos. Un retour en temps réel est fourni à l’utilisateur permettant de guider l’utilisateur tout au long de la reconstruction.

Une idée similaire est présente dans *SweepCanvas* de Li et al. [LLZ<sup>+</sup>17]. L’utilisateur dessine par-dessus une image RGB-D de manière à produire rapidement des prototypes 3D dans la scène représentée par l’image RGB-D. En utilisant deux courbes, l’utilisateur peut créer facilement des surfaces 3D, appelées *toiles de balayage*, en se basant sur les plans présents dans le contexte existant. L’avantage de cette méthode est que contrairement à la précédente, les surfaces créées n’ont pas besoin d’être elles-mêmes planaires. Cependant elle ne reconstruit pas non plus un véritable maillage 3D pour chaque élément dessiné.

L’utilisation d’un modèle 3D pour aider à la reconstruction a également inspiré des méthodes dédiées à la reconstruction de personnages. *ArtiSketch* de Levi et Gotsman [LG13] et *Modeling Character Canvases from Cartoon Drawings* de Bessmeltsev et al. [BCV<sup>+</sup>15] utilisent toutes les deux des squelettes 3D, placés dans la même position que celle du personnage esquissé, de manière à créer un maillage articulé correspondant au croquis de l’utilisateur en approximant les parties du corps du personnage par un cylindre généralisé (fig. 2.29).

*ArtiSketch* nécessite des données en entrée bien précises. Il est nécessaire de fournir plusieurs dessins de la silhouette du personnage sous plusieurs poses différentes, en indiquant en plus la correspondance entre chaque morceau du corps de la silhouette et la partie du squelette correspondante. Le système se sert ensuite de la pose de chaque

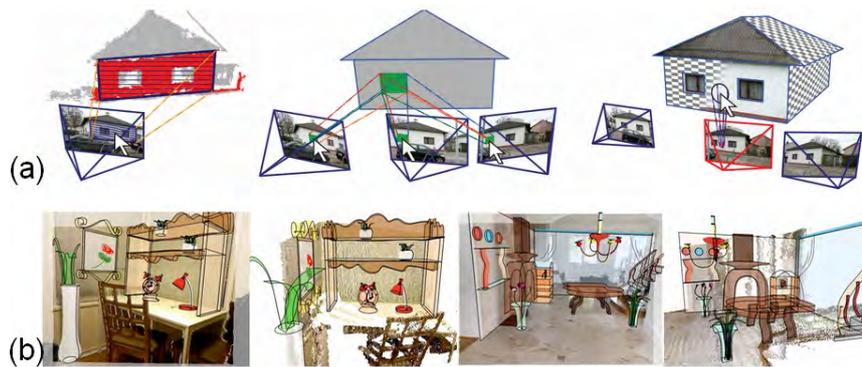


FIGURE 2.28 – Utilisation de données brutes comme support pour la reconstruction. Un nuage de points obtenu à partir de multiples photos est utilisé pour la reconstruction de bâtiments avec *Sketch-based guided modeling of 3D buildings from oriented photos* [SKMW17](a), des données RGB-D sont utilisées pour définir le contexte dans lequel l'utilisateur peut créer des surfaces 3D avec *SweepCanvas* [LLZ<sup>+</sup>17] (b).

squelette 3D pour déterminer l'angle de vue de la caméra correspondante de manière à effectuer un calibrage de celle-ci. Il se sert ensuite des informations déduites de ce calibrage et de celles fournies par l'utilisateur pour effectuer la reconstruction du personnage en 3D.

À l'inverse *Modeling Character Canvases from Cartoon Drawings* calcule la correspondance entre le dessin et le squelette de manière automatique, et ne nécessite qu'un dessin en entrée, selon un seul point de vue. Cette correspondance est effectuée en segmentant le contour dessiné par l'utilisateur de manière à ce que cette segmentation corresponde à la topologie du squelette et reflète la proximité des os dans celui-ci. Par la suite, la position 3D des courbes de la silhouette est calculée et les surfaces de chaque membre sont créées et assemblées pour former le modèle final.

Le défaut principal de ces deux méthodes repose cependant sur la nécessité de fournir en entrée un squelette 3D dans la pose correspondant au dessin, signifiant que l'utilisateur doit avoir des connaissances dans la manipulation de squelette 3D pour arriver à le positionner comme il faut. Il est également nécessaire que les proportions du dessin correspondent à celles du squelette.

Plus généralement, le défaut des méthodes nécessitant un modèle 3D comme support est que l'utilisateur ne peut pas créer un nouveau modèle à partir de rien, ce qui limite l'utilisation de ce genre d'applications.

## 2.7 Réseaux de neurones

Pour finir, un nouveau domaine de recherche a pris de l'ampleur au cours de ces dernières années : les réseaux de neurones. Ces méthodes basées sur l'apprentissage,

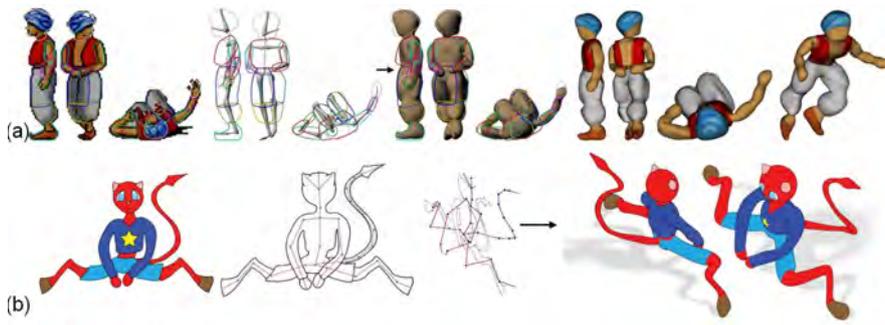


FIGURE 2.29 – Deux méthodes utilisant un squelette 3D avec la pose correspondante à celle du personnage dessiné comme données d’entrée : (a) *ArtiSketch* [LG13] qui utilise plusieurs vues de la silhouette et (b) *Modeling Character Canvases from Cartoon Drawings* [BCV<sup>+</sup>15] qui fonctionne avec une seule.

en utilisant une structure CNN (*Convolutional Neural Network*), ont démontré de très bonnes performances dans de nombreux domaines, y compris pour la reconstruction 3D à partir de croquis. Pour résoudre l’ambiguïté inhérente au passage de la 2D à la 3D, les réseaux de neurones sont entraînés sur des millions de données composées des croquis et des formes 3D correspondantes. La différenciation entre les méthodes porte principalement sur la stratégie adoptée : quels sont le type de données apprises et le type de données produites ? En effet, il n’est pas simple de définir *ce que doit apprendre* le réseau pour être capable d’effectuer une reconstruction 3D à partir d’un croquis.

Certaines méthodes ont donc décidé de travailler sur des modèles paramétrables.

*DeepSketch2Face* de Han et al. [HGY17] permet la création rapide d’un visage 3D caricatural (fig. 2.30). Les auteurs ont commencé par créer un modèle 3D déformable bilinéaire d’un visage permettant de représenter l’espace défini par l’ensemble des différents types de visages et des expressions présents dans leurs bases de données. Ils ont ainsi créé une correspondance directe entre les paramètres du modèle et la forme et l’expression du visage. Par la suite, le réseau est entraîné avec des dessins générés à partir des modèles de leur base, de manière à être capable d’identifier la forme du visage et l’expression qu’elle représente. Une fois fait, l’utilisateur peut tracer la silhouette d’un visage, ainsi que les lignes caractéristiques définissant l’expression qu’il désire avoir. Le réseau de neurones sera alors en mesure de déterminer les paramètres nécessaires pour modifier le modèle 3D déformable de manière à correspondre au croquis de l’utilisateur.

Cette idée de modèle paramétrique déformable se retrouve également chez *Deep Sketch-Based Modeling of Man-Made Shapes* de Smirnov et al. [SBS19] qui est dédié à la reconstruction d’objets manufacturés. Leur système se base sur un modèle paramétrique déformable composé de patches de Coons, c’est-à-dire de surfaces fermées et délimitées par 4 bordures qui partagent leurs points finaux. Il connaît, pour chaque catégorie d’objet qu’il peut reconstruire, un modèle référent topologiquement simple qui

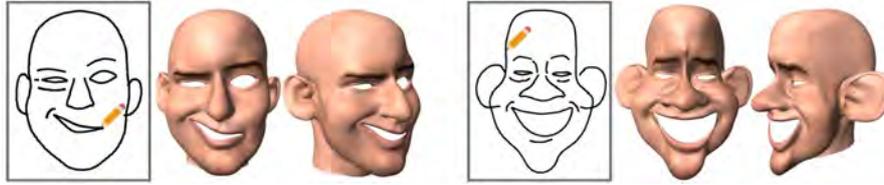


FIGURE 2.30 – Visages caricaturaux créé à partir d’un croquis à l’aide du réseau de neurones *DeepSketch2Face* [HGY17].

est utilisé pour structurer la reconstruction. Le réseau est alors entraîné en recevant en entrée un modèle 3D et le dessin correspondant, généré à partir du modèle et légèrement modifié de manière à obtenir un rendu identique à celui d’une véritable esquisse. Il cherche alors à produire un ensemble de patches de Coons et à les assembler pour former la forme finale, correspondant au modèle 3D en entrée, tout en prenant en considération la structure du modèle référent correspondant à la catégorie de l’objet.

*Shape Synthesis from Sketches via Procedural Models and Convolutional Networks* de Huang et al. [HKYM17] fonctionne sur une idée similaire, et présente une architecture de réseau permettant l’apprentissage de multiples paramètres de modèles procéduraux. Le réseau est entraîné en fournissant en entrée un dessin et les paramètres pour obtenir le modèle correspondant. Leur système est donc capable de modéliser n’importe quel modèle procédural, à condition de l’entraîner sur ce type de formes avant. Il fonctionne donc très bien pour des objets tels que des arbres, des bijoux ou des récipients comme montrés dans leur article ainsi qu’à la fig. 2.31.

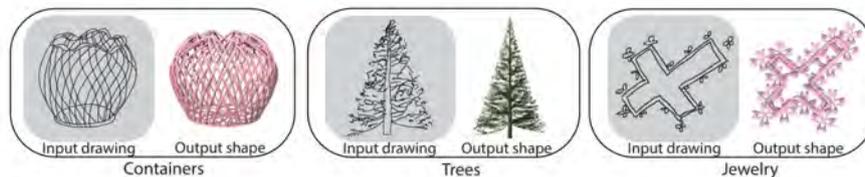


FIGURE 2.31 – Quelques résultats obtenus par *Shape Synthesis from Sketches via Procedural Models and Convolutional Networks* [HKYM17] le réseau de neurones spécialisé dans les modèles procéduraux.

*AtlasNet* de Groueix et al. [GFK<sup>+</sup>18] est une méthode qui va représenter la forme 3D comme étant un ensemble de surfaces paramétriques. Ce réseau est capable de produire à partir d’une seule image 2D, ou d’un nuage de points 3D, le maillage triangulaire 3D correspondant ainsi qu’un Atlas des surfaces paramétriques pouvant être utilisées pour texturer le modèle ou effectuer son impression 3D. Cette méthode repose sur la recherche de paramètres qui permettent de transformer un ensemble de carrés 2D et de les positionner en 3D pour obtenir la forme finale, de la même manière que l’on combine

des bandes de papiers pour réaliser une sculpture en papier mâché. Ce réseau est entraîné sur des paires images-maillage 3D et nuages de point-maillage 3D, comme il accepte les deux types de données en entrée.

L'un des points forts de cette méthode est qu'elle est capable de produire un résultat y compris pour des modèles non présents dans les données d'entraînement, bien que ceux-ci présenteront quelques artéfacts (fig. 2.32). De plus, elle permet la reconstruction de modèles complexes avec une topologie différente d'un disque. Cependant, le choix du nombre de paramètres à utiliser pour représenter la forme est difficile et n'est pas automatique. S'il n'est pas assez élevé, les triangles du maillage peuvent être déformés, et s'il y en a trop les surfaces paramétriques peuvent se chevaucher ou ne pas se toucher du tout.

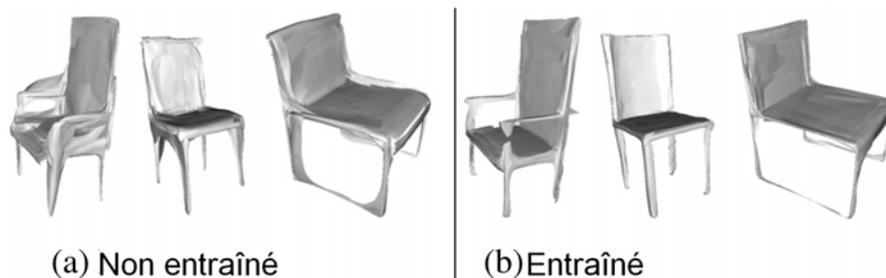


FIGURE 2.32 – *AtlasNet* [GFK<sup>+</sup>18] est capable de produire un résultat même lorsque le réseau n'a pas été entraîné sur le type d'objet à reconstruire (a), même si ces résultats sont de moins bonne qualité que lorsque l'entraînement contenait la catégorie d'objet (b).

D'autres méthodes ont décidé de travailler à partir de multiples vues du dessin, puisque cela fournit plus d'informations pour guider la reconstruction 3D.

*3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks* de Lun et al. [LGK<sup>+</sup>17] est un réseau représentant la profondeur de la surface et les normales sur plusieurs points de vue. Ces cartes de profondeurs sont ensuite fusionnées en un nuage de points 3D, qui est à son tour converti en un polygone. Le réseau est entraîné de manière à produire des cartes de normales et de profondeurs selon 12 points de vue différents à partir d'une image d'un seul point de vue spécifique (de face, de côté, de haut) ou depuis plusieurs dessins représentant la même forme depuis des vues différentes. Il effectue donc son apprentissage sur des données composées de dessins, générés à partir d'un modèle 3D, ainsi que des 12 cartes de profondeurs et de normales correspondant à chacun. Notons que leur réseau est décomposé en deux parties. La première sert à fournir une version encodée du dessin représentant la forme dessinée, et la deuxième qui prend cette version encodée en entrée afin de produire les cartes de profondeurs et de normales. On peut donc facilement changer la deuxième partie pour changer le but du réseau, en utilisant par exemple uniquement l'encodeur pour effectuer une reconnaissance de forme.

Des exemples de résultats produits avec ce système sont montrés fig. 2.33.

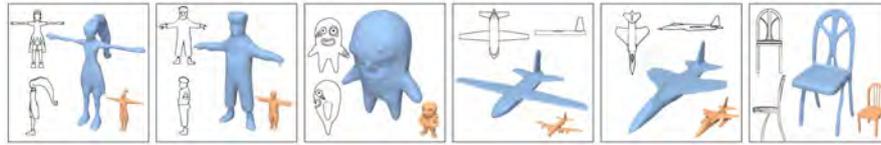


FIGURE 2.33 – Modèles reconstruits à partir de croquis sous 2 angles de vues à l’aide du réseau de neurones *3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks* [LGK<sup>+</sup>17]. Le modèle orange est le modèle 3D le plus proche du dessin dans les données d’entraînement du réseau.

Le principe de fusionner le résultat de différentes vues pour obtenir le modèle final se retrouve également chez *Robust Flow-Guided Neural Prediction for Sketch-Based Freeform Surface Modeling* de Li et al. [LPL<sup>+</sup>18]. Leur système se décompose en deux parties. Un premier réseau va générer un champ d’écoulement, représentant le flux de courbures sur toute la surface, à partir de l’esquisse de l’objet selon l’angle de vue choisi par l’utilisateur et du masque binaire correspondant, calculé automatiquement. Pour ce faire, il est demandé à l’utilisateur de faire manuellement la distinction entre trois types de traits qui composent le dessin : les contours, les lignes de pliage et les lignes nets. Le second réseau va ensuite utiliser le dessin et le champ d’écoulement afin de générer la surface 3D pour cette vue, accompagné d’une carte de profondeur et de normale décrivant la surface. Ces informations, obtenues pour de multiples vues, servent ensuite à fusionner les surfaces et obtenir le modèle final.

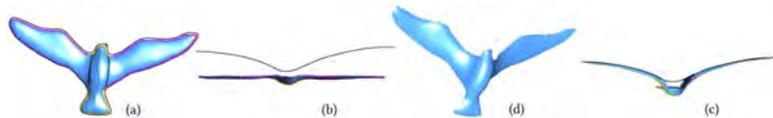


FIGURE 2.34 – Exemple du processus de reconstruction d’un oiseau avec *Robust Flow-Guided Neural Prediction for Sketch-Based Freeform Surface Modeling*[LPL<sup>+</sup>18]. En dessinant la forme sous un nouvel angle de vue (b), l’utilisateur apporte des informations supplémentaires qui permettent de modifier la surface générée avec le premier sketch (a) afin d’obtenir la forme finale (c)(d).

*What You Sketch Is What You Get* de Delanoy et al. [DAI<sup>+</sup>18] se sert également de dessins sous de multiples vues. Cependant au lieu de fournir tous les dessins en entrées, ceux-ci sont fournis successivement de manière à mettre à jour le modèle reconstruit. Leur méthode se sert d’un premier croquis en perspective pour prédire une première reconstruction de l’objet volumétrique, qui peut ensuite être affinée par l’utilisateur, ou même compléter avec de nouveaux éléments, en fournissant des esquisses supplémentaires selon d’autres points de vue (fig. 2.35). Leur système se compose donc en réalité de deux

réseaux : le premier prend en entrée un dessin unique, et est entraîné pour en proposer une reconstruction voxelisée ; le second est le réseau chargé d'affiner le premier résultat de manière itérative, il prend donc en entrée un dessin et un objet volumétrique déjà existant et doit modifier celui-ci pour qu'il corresponde au nouveau croquis. Pour entraîner ces réseaux, il est donc nécessaire d'avoir des paires de dessins et d'objets volumétriques correspondants, et si possibles, que ces paires correspondent aux objets pour lesquels ce réseau sera susceptible d'être utilisé par la suite.

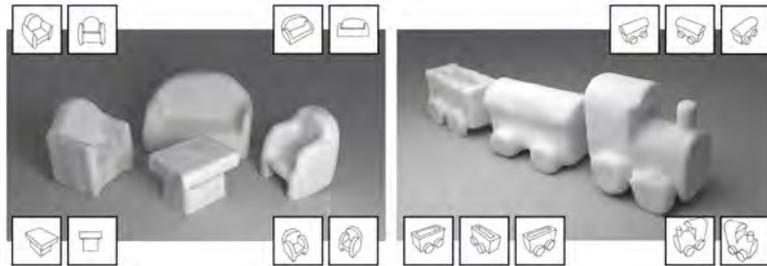


FIGURE 2.35 – Deux scènes modélisées avec *What You Sketch Is What You Get*[DAI<sup>+</sup>18]. Chaque modèle présent dans la scène a été reconstruit à l'aide de deux ou trois croquis montrés dans les encarts.

Bien que ces méthodes basées sur CNN montrent des résultats prometteurs, il reste encore beaucoup à effectuer dans ce domaine. La collecte des données de création, ainsi que l'entraînement des réseaux restent des tâches qui demandent beaucoup de temps. De plus, les réseaux de neurones ne marchent vraiment bien que dans les catégories correspondant à celles de leurs données d'entraînement. Si on leur présente des dessins non conformes à leurs apprentissages, ils seront dans l'incapacité de fournir un résultat de qualité. De plus, il semble encore impossible à l'heure actuelle de créer avec ces méthodes des formes plus complexes et sophistiquées, telles que des formes superposées ou avec un genre supérieur à 1.

## 2.8 Récapitulatif

Un récapitulatif des articles présentés dans cette section est disponible sous la forme de deux tableaux fig. 2.36 et fig. 2.37. Le premier tableau indique pour chacune de ces méthodes la nature des données que l'utilisateur doit fournir afin d'obtenir un résultat ainsi que le type de méthode, c'est-à-dire si celle-ci est complètement automatique, incrémentale ( l'utilisateur dessine les éléments les uns après les autres pour les ajouter au modèle 3D et guider la reconstruction) ou interactive (l'utilisateur doit utiliser des widgets pour guider la reconstruction). Le second tableau présente le type de résultats que chacune de ces méthodes permet de produire ainsi que la nature des objets reconstituables. L'information de temps est également indiquée. Notez cependant que le temps

requis pour les méthodes pleinement automatiques ne correspond qu'au temps nécessaire pour effectuer la modélisation une fois le dessin fourni, et est de ce fait forcément plus court que le temps nécessaire pour les méthodes qui sont interactives, et donc nécessitent que l'utilisateur guide la reconstruction, ou incrémentales, qui prennent en compte le temps que l'utilisateur passe à dessiner toutes les parties du modèle. De plus, lorsque les données sur le temps nécessaire ne sont pas fournies dans l'article ceci est indiqué comme non disponible par l'abréviation *n.d.* dans le tableau.

	Données en entrée					Méthode			
	Un seul croquis	Un croquis et des annotations	Plusieurs croquis	Modèle 3D	Squelette 3D	Nuage de points 3D	Incrémentale	Interactive	Automatique
Sketch [ZH96]		X					X		
GiDeS [PJB00]		X					X		
Structured Annotations for 2D-to-3D modeling [GI209]		X						X	
BendSketch [LPL+17]		X						X	
Ink-and-Ray [SKV+14]		X							X
Interactive High-Relief Reconstruction for Organic and Double-Sided Objects from a Photo [YHU+17]	X							X	
Seamless Reconstruction of Part-Based High-Relief Models from Hand-Drawn Images [DNJ+18]		X							X
Sketch-Based 3D Hair Posing by Contour Drawing [SI17]	X			X					X
Gesture3D [BVS16]	X				X				X
Sketch Abstractions for Character Posing [HMC+15]	X			X					X
Differential Blending for Expressive Sketch-Based Posing [OBP+13]	X			X	X			X	
The Line of Action: an Intuitive Interface for Expressive Character Posing [GCR13]	X			X					X
SketchMo: Sketch-based Motion Editing for Articulated Characters [CIRI+16]	X			X				X	
3D modeling with silhouettes [RDI10]			X				X		
Progressive sketching with instant previewing [WZS19]			X				X		
ShapeShop [SWSJ07]			X				X		
FiberMesh [NISHA07]			X				X		
ILoveSketch [BBS08]			X				X		
Analytic Drawing of 3D Scaffolds [SKSK09]			X				X		
BendFields [BB15]	X								X
True2Form [XCS+14]			X				X		
SymmSketch [MHZ+15]			X				X		
Modeling 3D animals from a side-view sketch [EBC+15]	X								X
RigMesh [BJD+12]			X				X		
SmoothSketch [KH06]	X								X
Free-Form Sketching of Self-Occluding Objects [CSPyN11]	X								X
Floral diagrams and inflorescences [IOI05]			X				X		
3D layered articulated object from a single 2D drawing [No+17]		X							X
Automatic Single-View Character Model Reconstruction [BMD13]	X								X
Automatic structuring of organic shapes from a single drawing [EPB+19]	X								X
MotionMaster [LGY+06]			X						X
Efficient Sketch-Based Creation of Detailed Character Models through Data-Driven Mesh Deformations [KYY+15]	X								X
Model-driven Sketch Reconstruction with Structure-oriented Retrieval [LHZ+16]		X							X
Geosemantic Snapping for Sketch-Based Modeling [SAG+13]	X							X	
Interactive sketching of urban procedural models [NGDA+16]			X				X		
Sketch-to-Design [XXM+13]			X				X		
Modeling By Example [FKS+04]			X				X		
Sketch-Based Search and Composition of 3D Models [LF08]			X				X		
MagicCanvas [SID7]			X				X		
Sketch2Scene [XCF+13]			X				X		
SecondSkin [PS15]			X	X			X		
skippy [KYC+17]	X			X					X
Model-guided 3d sketching [XFZ+19]			X	X			X		
Sketch-based guided modeling of 3D buildings from oriented photos [SKMW17]			X			X	X		
SweepCanvas [LLZ+17]			X			X	X		
ArtiSketch [LG13]			X		X				X
Modeling Character Canvases from Cartoon Drawings [BCV+15]	X				X				X
DeepSketch2Face [HGY17]	X							X	
Deep Sketch-Based Modeling of Man-Made Shapes [SBS19]	X								X
Shape Synthesis from Sketches via Procedural Models and Convolutional Networks [HKYM17]	X								X
AtlasNet [GFK+18]	X								X
3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks [LGK+17]			X						X
Robust Flow-Guided Neural Prediction for Sketch-Based Freeform Surface Modeling [LPL+18]			X				X		
What you Sketch is What You Get [GISW19]			X				X		

FIGURE 2.36 – Tableau récapitulatif des données d'entrées et du type de méthode.

	Résultat					Variété d'objets reconstruisibles				Temps requis						
	Courbes 3D	Relief	Maillage 3D	Surfaces 3D	Voxels	Pose 3D d'un maillage	Organique	Artificielle	Restricté à une catégorie spécifique	Restricté à des géométries spécifiques	Libre	< 1 min	< 5 min	< 10 min	< 30 min	> 30 min
Sketch [ZHK06]			X							X		n.d.	n.d.	n.d.	n.d.	n.d.
GoDe5 [P-BW05]			X							X		n.d.	n.d.	n.d.	n.d.	n.d.
Structured Animations for 2D-to-3D modeling [G1209]			X				X					n.d.	n.d.	n.d.	n.d.	n.d.
BendSketch [LPL+17]			X	X							X					X
Ink-and-Ray [SKV+14]		X					X					X				
Interactive High-Relief Reconstruction for Organic and Double-Sided Objects from a Photo [YH+17]		X					X									X
Seamless Reconstruction of Part-Based High-Relief Models from Hand-Drawn Images [DNI+14]		X					X					X				
Sketch-Based 3D Hair Posing by Contour Drawing [SH17]						X			X			X				
Gesture3D [SV+15]						X			X				X			
Sketch Abstractions for Character Posing [HMC+15]						X			X			X				
Differential Blending for Expressive Sketch-Based Posing [OBP+13]						X			X			X				
The Line of Action: an intuitive interface for Expressive Character Posing [GCR13]						X			X						X	
SketchMo: Sketch-based Motion Editing for Articulated Characters [CIRL+16]						X			X						X	
3D modeling with silhouettes [RDC10]			X					X								X
Progressive sketching with instant previewing [WZS15]			X								X	n.d.	n.d.	n.d.	n.d.	n.d.
ShapeShop [SWS+07]			X								X					X
FiberMesh [NCS+07]			X	X							X					X
LoVeSketch [BB08]		X						X								X
Analytic Drawing of 3D Scaffolds [SKSK09]		X						X								X
BendFields [HBB15]		X	X					X								X
True2form [XCS+14]		X						X								X
SymmsSketch [MHZ+15]			X							X	X					X
Modeling 3D animals from a side-view sketch [EBC+15]			X							X			X			
HighMesh [BJD+12]			X													X
SmoothSketch [KH06]			X								X	n.d.	n.d.	n.d.	n.d.	n.d.
Free-Form Sketching of Self-Occluding Objects [CSPyN11]			X							X		X				
Floral diagrams and inflorescences [HOH05]			X					X								X
3D layered articulated object from a single 2D drawing [No+17]			X					X				n.d.	n.d.	n.d.	n.d.	n.d.
Automatic single-view Character Model Reconstruction [BMOL13]			X					X				X				n.d.
Automatic structuring of organic shapes from a single drawing [EPB+19]			X				X						X			
MotionMaster [LDV+06]						X		X				X		X		
Efficient Sketch-Based Creation of Detailed Character Models through Data-Driven Mesh Deformations [KYY+15]			X						X			X				
Model-driven Sketch Reconstruction with Structure-oriented Retrieval [LHZ+16]				X				X				X				
Geometric Snapping for Sketch-Based Modeling [SAG+13]			X					X							X	
Interactive sketching of urban procedural models [NGDA+16]			X					X							X	
Sketch-to-Design [XOM+11]			X					X							X	
Modeling by Example [KSK08]			X								X					X
Sketch-Based Search and Composition of 3D Models [LF08]			X								X					X
MagicCanvas [SH07]			X					X							X	
Sketch2Scene [KCF+13]			X					X							X	
SacredSkin [PS15]			X					X							X	
Slappy [KYC+17]		X						X				X				
Model-guided 3d sketching [KFZ+19]		X						X							X	
Sketch-based guided modeling of 3D buildings from oriented photos [SKMW17]			X					X								X
SweepCanvas [LZ+17]				X				X								X
ArtSketch [LG13]			X					X				X				
Modeling Character Canvases from Cartoon Drawings [BCV+15]			X					X				X				
DeepSketch2Face [HCY17]			X					X							X	
Deep Sketch-Based Modeling of Man-Made Shapes [SB119]			X					X				X				
Shape Synthesis from Sketches via Procedural Models and Convolutional Networks [HKYM13]			X					X				X				
AtlasNet [GFX+18]			X								X	X				
3D Shape Reconstruction from Sketches via Multi-view Convolutional Networks [LGK+17]			X								X	X				
Robust Flow-Guided Neural Prediction for Sketch-Based Freeform Surface Modeling [LH+18]			X	X							X				X	
What you Sketch is What You Get [GISV19]					X						X				X	

FIGURE 2.37 – Tableau récapitulatif des résultats produits, de la variété d’objets qui peuvent être construits. L’information de temps est également indiquée. Pour les méthodes interactives et incrémentales, ce temps prend en compte l’intégralité du processus, dessin et interactions, alors que pour les méthodes automatiques il ne s’agit que du temps de reconstruction après que le dessin soit fourni. Lorsque l’information de temps n’est pas disponible, les cases sont remplies de *n.d.*.

## 2.9 Nos travaux

Les méthodes présentées dans ce chapitre permettent la reconstruction de modèles 3D grâce à des approches différentes. Cependant, bien que ces dernières permettent d’obtenir de bons résultats, quelques problèmes restent présents : les méthodes de reconstruction sont souvent limitées à un seul type d’objets, et certains objets ne disposent d’aucune

méthode adaptée à leur reconstruction ; de plus, certaines de ces méthodes nécessitent des données d'entrée (données 3D, dessins sous de multiples vues, etc.) ou un grand nombre d'interactions complexes ne leur permettant pas d'être accessibles à tous, ce qui est pourtant l'un des objectifs de la modélisation par croquis. Le but de cette thèse est d'apporter des solutions à ces problèmes.

Pour garder la simplicité que doivent avoir les méthodes de reconstruction par croquis, nos travaux ne requièrent qu'un seul croquis en entrée. Comme présenté section 1.4.1 notre premier système est dédié à la reconstruction d'objets floraux et est motivé par la nécessité croissante de pouvoir peupler aisément la flore des mondes virtuels. Plus précisément, notre objectif est de proposer une solution qui comble le défaut principal de la méthode *Floral diagrams and inflorescences* de Ijiri et al. [IOOI05], à savoir le temps nécessaire à la reconstruction. Leur système étant pleinement interactif avec l'utilisateur, la modélisation d'une fleur prend plusieurs dizaines de minutes, et reste donc un processus requérant beaucoup de temps et d'effort. Il n'existe aucune méthode qui permet la reconstruction de fleurs à partir d'un seul croquis, de manière simple et rapide. Nous avons donc cherché une solution permettant de minimiser les interactions requises avec l'utilisateur afin d'automatiser au maximum le processus de reconstruction, permettant ainsi à l'utilisateur d'obtenir son modèle de manière instantanée et sans effort. Puisque notre modéleur est limité à la reconstruction de fleurs, nous avons analysé l'ensemble des a priori sur lesquels nous pouvons nous baser pour arriver à cette automatisation. Le résultat de cette recherche est présenté section 3.8.

La seconde partie de cette thèse a été dédiée à la recherche d'une solution à un problème plus générique : la possibilité de générer n'importe quelle forme 3D à partir d'un croquis d'un seul point de vue. En effet, la modélisation d'une forme libre à partir d'un seul dessin reste un problème ouvert, la majorité des travaux présentés précédemment étant limité dans la nature de l'objet à modéliser. Ceci s'explique par la difficulté à positionner dans un espace 3D les éléments esquissés selon un seul point de vue, principalement à cause des parties cachées. Ce positionnement ne peut être fait automatiquement que grâce à la compréhension de la nature de l'objet représentée par le croquis. Or, comprendre cette sémantique implique de se baser sur des connaissances a priori, ou d'émettre des hypothèses sur la forme dessinée, ce qui va par conséquent limiter le type d'objets restructuribles par la méthode. Pour pouvoir reconstruire tout type de forme, il est donc nécessaire que le système ne se base sur aucun type d'a priori pour la modélisation. Nous avons donc cherché un moyen pour obtenir les informations nécessaires à la reconstruction autrement, et avons décidé qu'être guidé par l'artiste était une bonne solution. Toute la problématique a donc été de mettre au point un système qui permet à l'utilisateur de fournir les informations nécessaires à la reconstruction de tous types de formes et que cela se fasse par le biais d'un nombre minimal d'interactions simples de manière à rester accessible pour tous. La solution que l'on propose est présentée en détail section 4.7.

## Chapitre 3

# Reconstruction 3D de Fleurs à partir de croquis

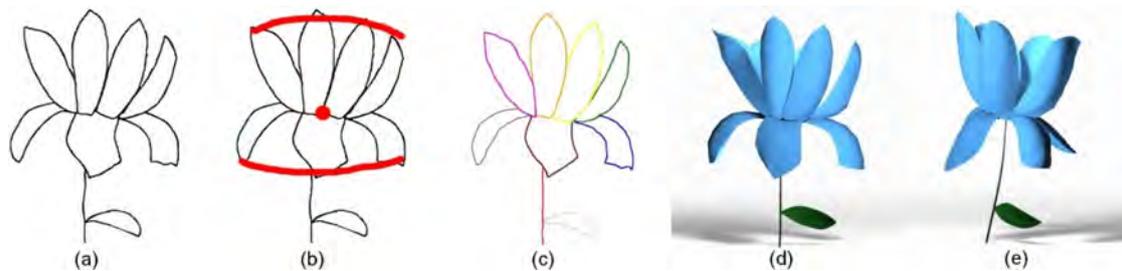


FIGURE 3.1 – Reconstruction d’un modèle de fleur à partir d’un croquis : le croquis d’entrée (a), les traits de guidage fournis par l’utilisateur qui seront utilisés pour la reconstruction du cône 3D (b), le croquis segmenté en pétales et autres éléments botaniques (c), modèle reconstruit du même point de vue que l’entrée (d) ; et d’un point de vue différent (e).

### 3.1 Introduction

Dans ce chapitre, nous décrivons notre première contribution, un nouveau système de reconstruction d’objets floraux à partir d’un croquis 2D dessiné par l’utilisateur. Ce système est le premier à permettre la reconstruction 3D de fleurs à partir d’un seul croquis, avec un point de vue arbitraire et d’une façon automatique. La méthode présentée dans cet article a fait l’objet d’un exposé à Pacific Graphics 2018 et d’une publication dans Computer Graphics Forum [BSCH18].

### 3.1.1 Problématique

Les fleurs qui nous entourent et leurs beautés ont toujours subjugué l'être humain, faisant de celles-ci l'objet de nombreuses représentations depuis le début de l'histoire de l'art. Cependant, à l'ère du virtuel, de simples images 2D ne sont hélas plus suffisantes pour subvenir à nos besoins. L'utilisation de modèles 3D de fleurs par les botanistes, par exemple, est de plus en plus courante. Cette nécessité d'avoir une représentation 3D est également présente chez les créateurs d'environnements virtuels désirant peupler la flore de leur monde d'une manière réaliste, c'est-à-dire avec de nombreuses espèces de fleurs. Cependant, la modélisation géométrique de tels modèles reste, aujourd'hui encore, une tâche très fastidieuse en raison de complexités structurelles et géométriques, ainsi que de l'énorme diversité que couvre le terme de *fleurs*.

Un des facteurs compliquant cette modélisation consiste dans le fait que les fleurs sont en réalité une entité obtenue par la combinaison de plusieurs éléments botaniques (pétales, feuilles, etc.) qui sont souvent très nombreux, certaines fleurs pouvant en contenir plusieurs dizaines. Il est donc nécessaire de faire attention à la forme de chaque élément et à l'impact que cette forme a sur la position de l'élément dans le modèle final. Ce point est d'autant plus important que la plupart d'entre eux possèdent des formes très minces et allongées nécessitant une interposition judicieuse les uns par rapport aux autres. Il faut également veiller à ce que chacun de ces éléments 3D reflète bien les détails caractéristiques qui lui sont propres afin de permettre une reconstruction réaliste. Par conséquent, un utilisateur expérimenté désirant créer un modèle 3D sur un modèleur classique doit être très soigneux lors de la modélisation, en prêtant attention à la fois à la structure hiérarchique des éléments et à leurs détails les plus fins.

Le second facteur rendant la modélisation géométrique complexe provient de la nature elle-même. La reconstruction d'une fleur est un processus dont les étapes vont varier que l'on doive reconstruire une marguerite, un lis glorieux ou une pivoine. Le terme *fleur* couvre une variété énorme d'espèces, chacune composée d'éléments avec des formes et des couleurs différentes, et cela sans même prendre en compte la possibilité qu'un artiste veuille modéliser une fleur totalement imaginaire. L'existence d'un intervalle de variations de formes si grandes rend impossible l'adaptation d'approches de reconstruction automatiques basées sur des données comme celles déjà appliquées avec succès à des visages humains, tel que [BV99], ou des corps, tel que [ACCP03] ou [SMT04].

Même en dehors de tout contexte de reconstruction, la modélisation 3D d'une fleur est une tâche ambitieuse, mais dans notre cas le problème est encore plus complexe. Puisque nous désirons une méthode permettant une reconstruction à partir d'un croquis, nous avons comme difficulté supplémentaire toute la problématique même de la reconstruction 3D à partir d'esquisse (section 1.4.2). Afin de proposer une reconstruction 3D réaliste et facile, accessible à tous les utilisateurs, notre système doit donc :

1. être capable d'identifier les éléments botaniques représentés dans le croquis composant la fleur dessinée par l'utilisateur ;
2. pouvoir proposer une modélisation géométrique 3D de chacun de ces éléments en conservant les détails propres à l'espèce de la fleur ;
3. déterminer automatiquement, à partir des informations 2D du dessin d'entrée, le positionnement, dans l'espace 3D, de chacun des éléments respectant à la fois le croquis de l'utilisateur et la structure hiérarchique des éléments composant une fleur ;

La solution que nous proposons repose sur deux constats. Le premier est que la plausibilité botanique d'une fleur est simplifiée lorsque l'utilisateur la dessine, car notre perception de celle-ci l'est aussi. Ainsi les dessins floraux simplifient souvent les détails géométriques et l'arrangement répété des pétales. Certaines parties botaniques, telles que le calice, sont même le plus souvent complètement omises. Par conséquent, nous pouvons nous contenter de modéliser uniquement cette version simplifiée de la fleur, telle qu'elle est représentée par l'utilisateur et qui sera suffisante pour obtenir un modèle réaliste, même si celui-ci ne sera alors pas botaniquement correct. Le second est que nous faisons l'hypothèse que les fleurs, même si elles appartiennent à des espèces différentes partagent une structure commune pouvant être approximée par une forme géométrique relativement simple, c'est-à-dire un cône (section 3.5). Cette idée n'est d'ailleurs pas nouvelle et a déjà été utilisée par [YGCO<sup>+</sup>14]. À l'aide de ces deux constats, nous avons conçu une méthode capable de produire des résultats fidèles à l'intention de l'utilisateur. Les modèles sont produits en utilisant une base de données d'éléments botaniques (section 3.3) dans laquelle sont récupérées les parties composant la fleur finale et auxquelles sont appliquées une déformation non rigide (section 3.6), .

### 3.1.2 Reconstruction de croquis guidée par des données 3D

Le système que nous avons mis au point permet la production d'un résultat en assemblant différents éléments 3D présents dans notre base de données. De ce fait, elle se situe dans la catégorie des *méthodes de reconstructions guidées par des données* (section 2.9), et est à la fois guidée par des a priori, puisque l'on sait quel est le type d'objet qui doit être reconstruit (une fleur), et par des modèles 3D, puisqu'elle s'appuie sur une base de données 3D préexistante. C'est donc par rapport aux méthodes de cette catégorie qu'il est nécessaire de se positionner précisément.

Dans notre cas, notre méthode n'est pas simplement guidée par des données 3D, mais les utilise également pour produire le résultat final. En effet, notre système utilise les éléments de la base comme composants du résultat final. Ces éléments 3D servent également comme apport en informations supplémentaires pour l'analyse du dessin, c'est-à-dire pour détecter le type botanique des éléments du dessin. En aucun cas les modèles 3D ne sont fournis par l'utilisateur comme données d'entrée supplémentaires, aux côtés du croquis, et/ou ne servent de support pour la reconstruction (en fournissant un contexte

3D permettant de comprendre les traits de l'utilisateur). De ce fait, notre méthode diffère déjà grandement de toutes celles qui utilisent des données 3D (ou RGB-D) comme support sur lequel effectuer la reconstruction ([KYC<sup>+</sup>17], [LLZ<sup>+</sup>17], [PS15], [SI17], [LG13] ou [BCV<sup>+</sup>15]).

Notre travail se rapproche plus de ceux qui, tout comme nous, utilisent des éléments 3D déjà existants afin de composer leur résultat. La philosophie générale de ces méthodes consiste à rassembler une vaste base de données de modèles 3D, des les découper et les combiner pour créer de nouveaux maillages correspondant aux attentes de l'utilisateur, et c'est exactement ce que fait notre modelleur. Des méthodes telles que *Sketch2Scene* [XCF<sup>+</sup>13] ou *MagicCanvas* [SI07] se concentrent sur la reconstruction d'une scène, et non d'un modèle 3D, ce qui n'est pas notre approche. À l'inverse *Geosemantic Snapping* [SAG<sup>+</sup>13], *Sketch-to-Design*[XXM<sup>+</sup>13], *Modeling By Example* [FKS<sup>+</sup>04] et *Sketch-Based Search and Composition of 3D Models* [LF08] vont permettre la reconstruction d'un modèle 3D dont l'ensemble des parties proviennent de modèles différents. Cette démarche est plus semblable à celle que nous avons adoptée.

Un point important et commun à toutes ces méthodes est que c'est l'utilisateur qui doit sélectionner manuellement le bon candidat parmi la liste de résultats retournée après la recherche dans la base de données. Il s'agit d'une étape que nous considérons comme lourde, et que nous désirons automatiser. De plus toutes ces méthodes fonctionnent de manière itérative. L'utilisateur ne fournit pas directement un croquis pleinement réalisé, mais va dessiner morceau par morceau en alternant les étapes d'esquisse et de modification du résultat 3D (sélection des candidats), nécessitant ainsi que l'utilisateur tourne le modèle 3D actuel pour dessiner selon un bon angle de vue. La tâche qui incombe à l'artiste n'est pas seulement la sélection manuelle des bons candidats, mais aussi les opérations de rotation dans l'espace 3D, ce qui écarte donc ces méthodes d'un *simple dessin*.

Nous avons conçu notre système de façon à éviter ces étapes supplémentaires. Le dessin est réalisé en une fois, et dans le domaine 2D uniquement, et la sélection des éléments 3D depuis la base de données est automatique. Le modelleur *Sketch2Scene* ([XCF<sup>+</sup>13]) est assez proche de notre méthode de ce point de vue, bien qu'il soit destiné à la reconstruction d'une scène et non d'un modèle 3D. Dans cette approche, comme dans la notre, le dessin se fait dans le plan  $Z=0$ , et la sélection des composants de la scène ainsi que leur positionnement est automatique. Cependant pour permettre cette automatisation leur système passe par une étape d'apprentissage. En recevant un ensemble de scènes 3D comme données, le modelleur apprend la relation sémantique entre les objets, c'est-à-dire leurs types et la position qu'ils ont entre eux. Bien que cela permette la reconstruction de manière autonome, cela introduit une faiblesse à leur méthode : si le dessin représente une scène avec une disposition différente de celles des données d'entraînement, et donc non conforme aux relations sémantiques apprises par le système, la reconstruction sera impossible. Il s'agit d'une problématique que nous n'avons pas dans notre cas, car la relation entre les éléments composants notre modèle final est directement déduite par

le dessin et les connaissances a priori de la structure botanique d'une fleur. Ainsi, aucune étape d'apprentissage sur des données 3D n'a été nécessaire et notre modéleur est capable de proposer une reconstruction tant que la fleur dessinée possède une structure botanique correcte.

### 3.1.3 Modeleurs floraux préexistants

Puisque le besoin en modèles 3D de fleurs est grandissant, de nombreux modéleurs dédiés spécifiquement à leur création ont vu le jour. Traditionnellement, les objets floraux sont modélisés procéduralement, à l'aide de systèmes tel que le système en L. Bien que ces méthodes à base de règles garantissent la création de modèles avec une structure botaniquement correcte, elles laissent peu de place à la créativité de l'artiste.

Les récentes techniques d'acquisition de forme d'objets réels sont également à l'origine de modéleurs floraux permettant la reconstruction pleinement automatisée et quasiment instantanée d'objets que l'on peut observer dans le monde réel. Récemment, Yan et al. [YGCO<sup>+</sup>14] ont montré comment reconstituer des modèles de fleurs texturés à partir d'une entrée de photo unique. Ils s'appuient sur des connaissances préalables concernant la structure de la fleur et l'aspect redondant des pétales pour tenter de reconstruire la fleur, y compris les parties obstruées. Cependant, leurs résultats se limitent aux fleurs largement ouvertes avec une vue presque de face, leur technique supposant une visibilité totale ou partielle de chaque pétale. Par contre, dans notre travail, les artistes sont libres d'utiliser la vue de leur choix pour esquisser la fleur. Zhang et al. [ZYFY14] ont adopté une approche basée sur la numération 3D. Un modèle déformable de pétale a été préalablement créé par numérisation de pétales de multiples fleurs. Puis ce modèle est utilisé pour la reconstruction de fleurs à partir d'un nuage de points 3D numérisé à partir d'une seule vue. Ijiri et al. [IYYI14] ont adopté la tomodynamométrie à rayons X pour la modélisation réaliste des fleurs. S'appuyant sur des données de volume sur la forme de la fleur, y compris la partie cachée, ils combinent des techniques interactives et variationnelles pour adapter les primitives géométriques d'arbre et de feuille aux régions de volume cible et effectuer la reconstruction. Bien que ces méthodes permettent une création d'un modèle 3D de manière quasi instantanée, il est nécessaire d'avoir l'objet réel sous la main au moment de la conception. De plus, ces méthodes n'offrent aucun contrôle sur le résultat produit et ne permettent donc pas de créer des fleurs imaginaires.

Ces limitations ont été à l'origine d'un autre type de modéleur : les systèmes interactifs. Ces derniers permettent, via des interactions avec l'utilisateur, d'avoir un contrôle total sur le résultat produit, et donc de modéliser des fleurs imaginaires. Ijiri et al. [IOI06] ont mis au point un système permettant de concevoir une scène composée de plusieurs fleurs. L'utilisateur va assister le processus tout au long de la reconstruction en commençant par spécifier l'aspect général de la fleur, à l'aide d'une esquisse initiale, puis en fournissant plus de détails, par le biais d'esquisses plus précises des éléments du modèle final ou de diverses interactions. Notre approche, comme la leur, utilise une bibliothèque

de modèles 3D préalablement constituée afin de construire le modèle final. Cependant, alors que chez eux la sélection des éléments est faite manuellement par l'utilisateur, nous utilisons directement l'esquisse afin de choisir automatiquement les éléments de la base qui composeront le modèle final. Ijiri et al. [IOOI05] ont également développé un modèleur de fleurs interactif offrant une modélisation en deux étapes pour la reconstruction de modèles de fleurs botaniquement plausibles. L'utilisateur commence par esquisser la structure d'un modèle floral, en s'inspirant des diagrammes floraux et des inflorescences ([RDC10]). Ensuite, la géométrie des éléments floraux, ainsi que l'inflorescence du modèle, est créée à l'aide de traits 2D fournis par l'utilisateur dans leur éditeur de géométrie. Ces systèmes requièrent donc de nombreuses interactions avec l'utilisateur, ne rendant donc pas la création de modèles floraux aussi rapides qu'on le souhaiterait.

### 3.1.4 Contribution

. Notre méthode présente plusieurs avantages par rapport aux modèleurs dédiés aux objets floraux déjà existants. Tout d'abord, elle permet à l'utilisateur de créer rapidement des fleurs à partir d'une vue arbitraire et avec une interaction minimale, ne nécessitant que l'esquisse de leur silhouette et quelques traits de guidage fig. 3.1. Ensuite, s'appuyant sur une base de données préconstruite de modèles d'éléments botaniques, elle peut créer des objets floraux sophistiqués à partir de dessins simplifiés. Notre système peut notamment être utilisé pour générer des modèles de fleurs comportant plusieurs couches de pétales et même des fleurs avec différentes formes de pétales (repliés sur eux-mêmes par exemple), sans être limité à la reconstruction de fleurs réelles.

En comparaison avec les méthodes de reconstruction fonctionnant par assemblage de données 3D préexistantes, nous avons réussi à mettre au point le système le plus automatisé possible en supprimant intégralement les interactions 3D nécessaires avec l'utilisateur, sans pour autant compromettre la qualité et la variété des reconstructions possibles.

À notre connaissance, nous sommes les premiers à proposer un système de modélisation 3D d'objets floraux à partir d'un dessin unique, sans restriction sur son point de vue. Notre système permet une reconstruction avec la rapidité et l'automatisme des méthodes de reconstruction par acquisition, tout en ayant la robustesse et la liberté de reconstruction des systèmes interactifs.

## 3.2 Aperçu

L'utilisateur fournit au modèleur un dessin de la silhouette d'une fleur, représentée sous l'angle de vue de son choix, afin de générer un maillage 3D correspondant à l'esquisse. Puisque notre méthode travaille à partir de croquis, la qualité artistique de l'entrée n'est pas fixe, et dépend de la capacité à dessiner de l'utilisateur. Nous supposons cependant que le croquis de la silhouette en entrée est réalisé avec un niveau d'abstraction modéré accessible à tout utilisateur, sans être trop simplifié pour autant. Ainsi les schémas trop abstraits ou à l'inverse les schémas trop détaillés, par exemple

les illustrations scientifiques avec des traits d'ombrage, ne sont pas gérés actuellement par notre méthode (fig. 3.2). La fig. 3.1 montre un exemple d'une esquisse typique d'une fleur, au niveau d'abstraction escomptée, qui est composée de plusieurs pétales dont la base est localisée autour du centre de la fleur.

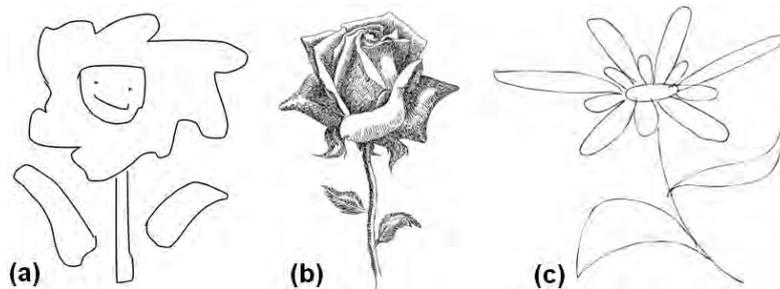


FIGURE 3.2 – Notre système ne fonctionne pas avec les dessins trop abstraits (a), ou trop détaillés (b), et requiert un niveau d'abstraction modéré (c).

Un point important que l'utilisateur doit respecter lorsqu'il nous fournit le croquis en entrée est que notre méthode suppose qu'une seule et unique fleur y est représentée. Notre système va dans un premier temps estimer les paramètres d'un ou de plusieurs *cônes 3D* représentant la géométrie globale de la corolle de la fleur. Cette tâche est accomplie grâce à des informations supplémentaires données par l'utilisateur sous la forme de *traits de guidage* dessinés interactivement sur l'esquisse de départ : un *centre de fleur*, c'est-à-dire le point de jonction entre la tige et la corolle, est donné via un simple clic, puis des courbes reliant les pointes des pétales présents sur le dessin, une pour chaque couche de pétales, et nommées *courbes d'extrémités* doivent être tracées. Un cône 3D, qui servira de base pour placer spatialement les éléments botaniques de la fleur lors de l'étape d'assemblage, est alors estimé pour chacune de ces courbes de guidage (section 3.5). Dans un second temps, le système analyse l'ensemble des traits constituant le dessin d'entrée, s'aidant du cône 3D ainsi que de certaines connaissances a priori sur les fleurs, afin de les segmenter en un ensemble de courbes élémentaires, chacune correspondant à un élément botanique tel que le pétale, la feuille et le centre de la fleur (l'ensemble d'étamines et du réceptacle floral, toujours simplifié sur les croquis et représenté comme un élément unique) (section 3.4). Les courbes élémentaires, obtenues par le biais de cette segmentation, sont utilisées comme données d'entrée pour des requêtes qui vont chercher et extraire d'une base de données, préalablement construite, les éléments botaniques 3D correspondant à ceux esquissés (section 3.3). Enfin, le modèle 3D final de la fleur est reconstruit en plaçant les éléments 3D récupérés autour du cône précédemment estimé, tout en respectant le croquis d'entrée et la structure botanique des fleurs (section 3.6). Une déformation non rigide est finalement appliquée à ces éléments afin d'obtenir une correspondance parfaite entre leurs silhouettes et le croquis d'entrée de l'utilisateur.

### 3.3 Construction d'une base de données de modèles floraux

Comme expliqué section 3.1.1, l'une des problématiques de la modélisation géométrique des fleurs consiste en l'existence d'une variété énorme de fleurs différentes, chacune composée d'éléments botaniques avec des détails propres à cette espèce. Notre modéleur se doit donc d'être capable de proposer une modélisation géométrique 3D de chacun de ces éléments qui conserve ces caractéristiques. Cependant, il s'agit d'une tâche extrêmement complexe. Afin de pouvoir modéliser un élément botanique possédant des caractéristiques propres à l'espèce de la fleur représentée sur le dessin, il faut être capable de reconnaître la nature botanique de l'élément en question (s'agit-il d'un pétale, d'une feuille, d'une tige, etc.), mais également l'espèce de la fleur du dessin, ou au moins l'espèce de la fleur dont est issu l'élément botanique dessiné. De plus, une fois identifiée il faudra encore savoir ce que cela entraîne ; notre modéleur ne doit pas se contenter de reconnaître, par exemple, un pétale de rose, il doit également savoir comment modéliser ce genre de pétale en 3D.

Cette nécessité de pouvoir identifier les éléments botaniques les uns des autres, de pouvoir identifier les espèces florales, et de savoir comment modéliser chacun de ces éléments par rapport à l'espèce en question indique qu'il est nécessaire que notre système possède un ensemble de connaissances a priori sur les éléments botaniques des différentes espèces de fleurs, de manière à être capable par la suite de les identifier sur un dessin, et d'en proposer un modèle 3D. Nous avons décidé de modéliser cet ensemble de connaissances sous la forme d'une base de données d'éléments botaniques 3D. Ce choix a été motivé par plusieurs raisons :

- La conception d'une base de données 3D n'est pas compliquée entre les modèles 3D déjà présents sur internet et la possibilité d'en créer nous-mêmes, à l'aide d'un logiciel de modélisation 3D classique tel que Maya et en utilisant des photos comme référence.
- Il est possible, dans une base de données, d'associer un ensemble de métadonnées à chaque maillage 3D, nous permettant ainsi d'avoir à la fois des informations sémantiques sur le modèle, telles que l'espèce de la fleur d'origine et le type de l'élément botanique (s'il s'agit d'une feuille, d'un pétale, etc.), ainsi que des informations sur sa structure 3D.
- Il existe déjà des méthodes dites de *Récupération basée sur l'esquisse* qui permettent de chercher dans une base de données de modèles 3D celui dont la projection 2D s'approcherait le plus du dessin utilisé comme donnée d'entrée pour la requête. Ces méthodes permettent donc d'avoir un lien direct entre une esquisse 2D et une base de données 3D.

La conception de la base de données a été effectuée en combinant des éléments botaniques réalisés par nous-mêmes et des éléments botaniques obtenus via internet. Notons que les modèles présents sur internet consistent principalement en des fleurs 3D com-

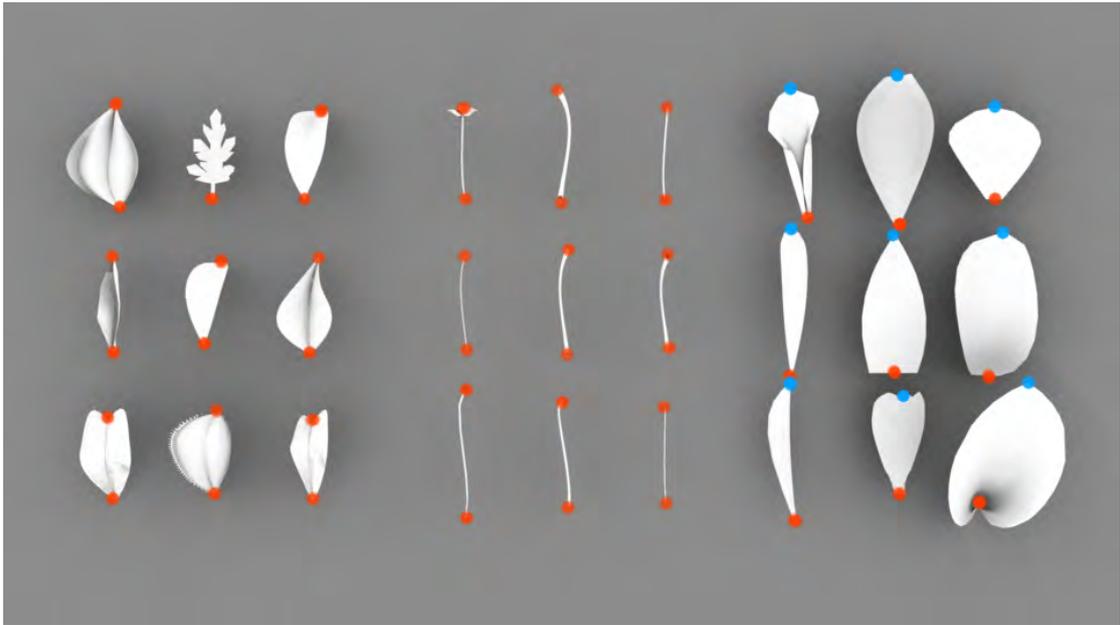


FIGURE 3.3 – Exemples de modèles d’éléments botaniques dans notre base de données : feuilles (à gauche), tiges (au milieu) et pétales (à droite). Les zones d’attachement sont indiquées en rouge et les extrémités des pétales en bleu.

plètes ; or nous désirons une base d’éléments botaniques individuels. Une étape de segmentation de ces modèles 3D en éléments botaniques a donc été réalisée manuellement, bien qu’une segmentation automatique et une reconnaissance des éléments puissent être appliquées à l’aide d’une technique d’apprentissage [KHS10]. Ainsi la base de données 3D utilisée dans le système actuel contient un ensemble de 86 modèles d’éléments botaniques obtenus via la segmentation de 20 modèles de fleurs de plusieurs espèces différentes (fig. 3.3).

De plus à chacun de ces modèles est associé un fichier de métadonnées, créé manuellement et contenant les informations nécessaires à notre programme, c’est-à-dire des informations sémantiques permettant l’identification de l’espèce de la fleur et du type botanique de l’élément, ainsi que des informations sur le maillage 3D qui nous seront utiles pour leur positionnement lors de l’étape de reconstruction. Un fichier de métadonnées contient donc les informations suivantes :

- Type botanique de l’élément (pétale, centre de la fleur, feuille, tige).
- Espèce de la fleur d’origine de cet élément (rose, marguerite, etc.)
- Zone d’attachement (l’ensemble des sommets du maillage pouvant être attachés à d’autres éléments 3D).
- Les points d’extrémités (les sommets du maillage 3D correspondant aux deux extrémités de l’élément botanique, c’est-à-dire la pointe du haut et de la base, à l’exception des centres de fleur où ce champ reste vide).

Chacun des modèles 3D de la base, étant associé avec un fichier de métadonnées, est alors labellisé en fonction de son type botanique. De plus, les profils spatiaux de chacun des éléments botaniques sont calculés et stockés à leur côté. Ces profils sont obtenus en effectuant de multiples projections 2D de la silhouette du maillage 3D selon un ensemble de différents points de vue, et seront utilisés pour effectuer la recherche dans la base de données à partir de croquis 2D.

### 3.4 Identification des éléments botaniques basée sur l'esquisse

Une fois que l'utilisateur a fourni le dessin de la fleur à notre système, ce dernier doit être capable de récupérer, depuis la base de données, l'ensemble de modèles 3D correspondant aux éléments présents dans le croquis d'entrée. Pour ce faire, il est nécessaire, dans un premier temps, d'effectuer une segmentation des courbes du croquis en courbes élémentaires, représentant chacune un élément botanique, c'est-à-dire un pétale, une tige, une feuille ou un centre de fleur. Par la suite, ces courbes élémentaires serviront de données d'entrée à une méthode de récupération basée sur l'esquisse, afin de récupérer de la base les modèles 3D correspondants qui seront utilisés pour la reconstruction du modèle final. Dans cette section, nous décrivons comment nous segmentons les courbes du croquis en courbes élémentaires (section 3.4.1) et les utilisons comme entrées pour la recherche de formes dans une base de données 3D (section 3.4.2).

#### 3.4.1 Segmentation du dessin

Une segmentation du croquis, et donc des courbes du dessin, en courbes élémentaires, consiste à identifier la ou les courbes représentant un même élément et à les séparer des autres courbes du croquis, puis à répéter ce processus jusqu'à que chacune de ces courbes appartienne à une courbe élémentaire. Ceci est effectué en détectant les limites des courbes du dessin, c'est-à-dire les points présentant une discontinuité spatiale ou temporelle par rapport aux autres courbes. Les intersections de deux courbes formant une jonction en T, ou encore la présence de cuspidés, représentent des discontinuités spatiales, et sont donc des candidats naturels pour ces limites de courbes. De même, les points correspondant au début ou la fin de chaque trait dessiné par l'utilisateur représentent une discontinuité temporelle utilisée pour la segmentation.

Un point important à prendre en considération pour la segmentation est que nous travaillons avec des dessins d'un niveau d'abstraction modéré. Pour ce genre d'esquisse, un élément botanique est dessiné, dans la plupart des cas, en un seul trait. Par conséquent, effectuer une segmentation des courbes en utilisant comme seul critère ses points de discontinuité est suffisant pour obtenir des courbes élémentaires. Cependant, l'utilisateur peut parfois utiliser plusieurs traits pour représenter un seul élément, il sera alors nécessaire de détecter que ceux-ci doivent être regroupés en une seule courbe élémentaire. Comme un niveau d'abstraction modéré est autorisé, l'inverse peut également être vrai,

et un seul trait de l'esquisse peut représenter plusieurs éléments. Il faudra alors scinder ce trait en plusieurs segments. Ces cas sont illustrés à la fig. 3.4(a-d). La fig. 3.4 (a,b) montre que deux traits représentent une flexion d'un même pétale et doivent donc être regroupés en un même élément botanique. A contrario, un seul trait sur la fig. 3.4 (d) doit être segmenté en plusieurs courbes, chacune représentant un pétale. Un autre cas à prendre en compte est pour la torsion d'un pétale. En effet, ce genre de torsion, comme illustrée à la fig. 3.4 (c), peut être dessinée en un ou deux traits. Lorsqu'il est dessiné en un seul trait, il faut être capable de le différencier du cas où un trait représente plusieurs pétales, comme illustrés en fig. 3.4 (d), et lorsqu'il est représenté en deux traits ceux-ci doivent alors être fusionnés un seul segment, comme dans les fig. 3.4 (a-b).

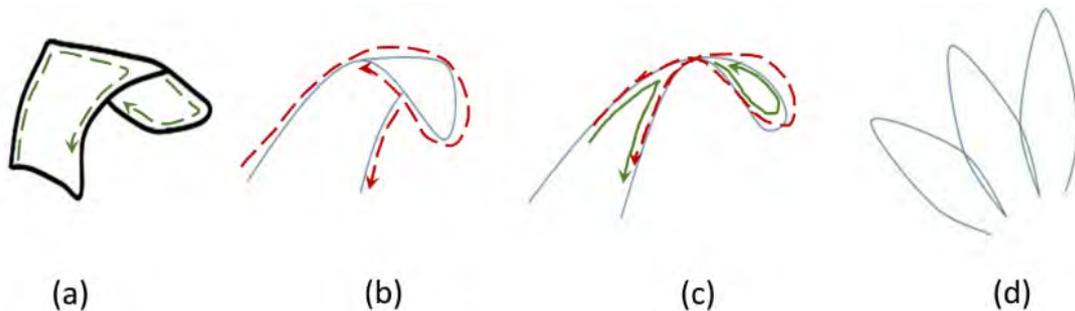


FIGURE 3.4 – Cas atypiques de fusion ou de scission d'un trait : deux traits représentent la courbure d'un pétale pouvant former un ou plusieurs points de jonction (a, b), la torsion d'un pétale peut être dessinée en un trait (rouge) ou en deux (vert) (c), un seul trait doit être segmenté en plusieurs courbes représentant chacune un pétale (d).

Ces cas particuliers sont gérés en utilisant deux principes simples qui prennent place après avoir effectué la segmentation des courbes selon leurs discontinuités.

Dans un premier temps nous vérifions pour chacune des courbes si elle contient une ou plusieurs boucles. Cette détection se fait simplement, puisqu'il suffit de vérifier s'il existe des points d'auto-intersection au sein de la courbe. (fig. 3.4(d) et fig. 3.4 (c) lorsque la courbe est tracée sous la forme d'un trait rouge unique). Si des boucles sont détectées, nous nous servons alors de l'aire à l'intérieur de celles-ci comme indication. Si l'aire est négligeable, on considère que la boucle n'est pas une caractéristique que l'utilisateur a sciemment mise dans le croquis, mais qu'il s'agit plutôt d'un artefact créé par celui-ci involontairement en dessinant rapidement, c'est-à-dire qu'il s'agit d'une boucle présente, car l'utilisateur a fait un « aller-retour » avec le stylo et que ce trait représente donc plusieurs éléments. Nous le segmentons alors au point de courbure maximale le long de la boucle (fig. 3.4 (d)). À l'inverse, si l'aire de cette zone est non négligeable, nous considérons alors qu'il s'agit d'un seul élément et ne segmentons pas davantage la courbe.

Deuxièmement nous cherchons les courbes qui doivent être fusionnées en un seul élément. Pour ce faire, nous regardons lorsque deux courbes successives dans le temps, c'est-à-dire qu'elles ont été dessinées à la suite, se rencontrent en un point de jonction.

Si cette jonction rentre dans un des deux cas suivants alors nous considérons qu'il est nécessaire de les fusionner en un seul élément :

1. Si l'une des deux courbes se termine par une jonction en T avec elle-même, et que l'angle entre les deux courbes est supérieur à un seuil de  $35^\circ$  (fig. 3.4 (b)).
2. Si des paires de directions tangentes se confondant peuvent être trouvées au niveau de la jonction entre les deux courbes (fig. 3.4 (c) quand elle est dessinée par deux traits verts).

Enfin, si les deux courbes successives dans le temps se rencontrent à deux points de jonction différents, nous vérifions si l'un des angles de jonction est supérieur au seuil de  $35^\circ$  (fig. 3.4 (a)). Le cas échéant, nous considérons qu'elles représentent en réalité différentes parties d'un même élément et qu'elles doivent alors être fusionnées.

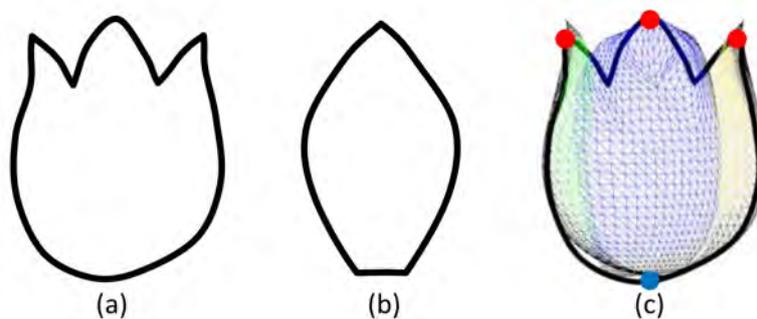


FIGURE 3.5 – Exemple de croquis qui ne peut pas être segmenté automatiquement (a). L'utilisateur est invité à dessiner un pétale référent(b), qui permet au système de récupérer la forme 3D correspondante. Ensuite, l'utilisateur localise les extrémités des pétales dans l'esquisse (points rouges (c)). Ces extrémités, ainsi que la pointe du cône (point bleu (c)), servent à placer les pétales sur le dessin (représentés en fil de fer de couleurs différentes).

**Dessin du pétale référent.** La segmentation du dessin en courbes élémentaires a pour objectif d'obtenir un ensemble de courbes 2D correspondant aux éléments botaniques qui composent la fleur dessinée par l'utilisateur. Ces courbes élémentaires sont par la suite utilisées par la méthode de récupération basée sur l'esquisse afin de récupérer depuis la base de données le maillage 3D et les informations correspondants à l'élément dessiné. Afin que la méthode de récupération fonctionne correctement, il est nécessaire que la courbe décrivant l'élément botanique soit un minimum détaillé. Lorsque l'esquisse est dessinée avec un niveau d'abstraction trop élevé, les méthodes de segmentations expliquées plus haut ne réussissent pas à proposer une segmentation en courbes élémentaires correctes. C'est par exemple le cas lorsque l'utilisateur dessine un ensemble de plusieurs pétales à l'aide d'une seule courbe, mais qu'aucune boucle n'est détectée (fig. 3.5(a)). Dans ce cas, le programme considèrera cet ensemble de pétales comme étant une courbe élémentaire, et l'utilisera comme donnée d'entrée pour la méthode de récupération qui

échouera par conséquent. Lorsque cela arrive, le système demande à l'utilisateur de dessiner une courbe représentant la vue de face d'un pétale que l'on appelle le *pétale référent* (fig. 3.5 (b)). C'est cette nouvelle courbe qui est alors utilisée comme donnée d'entrée pour la requête de récupération dans la base de données 3D, à la place de la courbe problématique du dessin d'origine. De plus, étant donné que l'échec de la segmentation de la courbe des pétales du dessin d'origine entraîne intrinsèquement l'échec de la détection automatique de la pointe du pétale sur l'ellipse 2D dans la partie suivante de notre méthode (section 3.6), l'utilisateur est invité à spécifier manuellement la location des extrémités de chacun des pétales en cliquant directement sur leur position sur le dessin d'origine (fig. 3.5 (c)). Notons qu'il peut être également demandé à l'utilisateur de dessiner un pétale référent lorsque la méthode de récupération basée sur l'esquisse ne trouve aucun élément botanique dans la base de données qui correspondrait à l'esquisse d'origine de l'utilisateur. Dans de tels cas, il serait informé par notre système de l'incapacité à trouver de bons modèles et invité à fournir un pétale référent qui servira de substitut pour une nouvelle recherche et pour la reconstruction 3D.

### 3.4.2 Identification des éléments botaniques

Une méthode de récupération basée sur l'esquisse peut alors être utilisée afin de récupérer depuis notre base de données 3D les éléments botaniques 3D et les informations correspondant aux courbes élémentaires obtenues par segmentation. Dans notre système cette méthode est une implémentation de [ERB<sup>+</sup>12]. Comparé à d'autres méthodes de ce genre (tel que [SXY<sup>+</sup>11]), celle-ci a l'avantage de permettre une correspondance partielle entre le dessin d'entrée et les images obtenues par rendu, ce qui est nécessaire dans notre application où les pétales du croquis sont généralement cachés par d'autres éléments. De ce fait, seule la partie visible du pétale, c'est-à-dire la partie dessinée, peut être utilisée pour la récupération du modèle 3D.

L'idée principale derrière cette méthode est qu'il est plus simple d'effectuer une comparaison entre deux images, et de trouver des images ressemblantes, plutôt que de comparer une image 2D et un maillage 3D. La première étape consiste donc à générer, pour chaque modèle 3D présent dans la base de données, un ensemble d'images correspondant à leur projection 2D selon différents points de vue.

Une fois que le problème de comparaison est exprimé dans le domaine 2D (comparaison image à image), il est possible d'utiliser les méthodes de recherche qui se basent sur le modèle *bag-of-features* [SZ03]. Le programme va dans un premier temps chercher l'intégralité des caractéristiques qui représentent les images obtenues dans l'étape précédente, exprimant ainsi chacune des projections 2D sous la forme d'un ensemble de caractéristiques locales.

Puisque cette détection de multiples caractéristiques est effectuée sur toutes les projections 2D issues de chacun des modèles 3D de la base, le nombre total de caractéristiques obtenues est très élevé (plusieurs millions) et elles sont toutes considérées comme différentes les unes des autres, ce qui ne permet pas d'effectuer des comparaisons entre

elles. Il est donc nécessaire d'ajouter une étape de classification qui va chercher à regrouper les caractéristiques similaires entre elles afin de déterminer une représentation générique de ces caractéristiques. Cette étape est effectuée par le biais d'un regroupement en k-moyenne [Llo82] sur un échantillon d'un million de caractéristiques sélectionnées uniformément sur l'ensemble des projections 2D des modèles 3D de la base. L'ensemble des centroïdes du regroupement résultants forme le *vocabulaire visuel* de 2500 mots où chaque centroïde, correspondant à mot visuel, représente les caractéristiques locales de son groupe.

Ensuite, le programme va exprimer chacune des projections 2D sous la forme d'un histogramme de fréquences de mots visuels. Pour ce faire, il va chercher pour chaque caractéristique de l'image le mot visuel le plus proche dans le vocabulaire visuel précédemment défini, puis va compter le nombre d'occurrences de chaque mot utilisé pour représenter l'image afin d'établir l'histogramme.

Cette étape de génération d'un vocabulaire visuel et d'expression des projections selon celui-ci n'est effectuée qu'une seule fois. Par la suite, lorsque l'utilisateur effectuera une recherche dans la base de données en fournissant un croquis, la courbe élémentaire dans notre cas, le programme va convertir celui-ci en histogramme de fréquences de mots visuels, par le biais du même procédé de détection de *bag-of-features*, puis va chercher dans la base de données les projections 2D avec les histogrammes dont la similarité, notée  $S_{similarity}$ , est la plus grande. Une fois trouvées, il retournera les modèles 3D correspondants à ces projections accompagnés de l'angle de vue selon lequel celles-ci ont été générées. Pour plus de détails sur la génération du vocabulaire et des histogrammes, j'invite les lecteurs à se référer au papier d'origine[ERB<sup>+</sup>12].

Notons que nous avons décidé d'utiliser 117 points de vue différents pour la première étape de la conception de notre base de données, qui a donc été de générer, pour chacun des modèles 3D, un ensemble d'images correspondant à leur projection 2D selon différents points de vue. 102 d'entre eux correspondent à des points de vue répartis uniformément sur la sphère de vue, les 15 restants correspondant à des vues presque frontales. De cette façon, nous nous assurons d'avoir une représentation 2D de chacun des éléments sur un intervalle d'angle de vue qui couvre l'ensemble des points de vue que l'utilisateur peut choisir, en générant davantage de vues presque-frontales, car les artistes ont tendance à dessiner principalement sous ce genre d'angle de vue.

Ainsi, une fois que le croquis a été segmenté en courbes élémentaires, notre système va chercher dans la base les modèles 3D correspondants. Pour ce faire, et comme expliquer précédemment, il commence par représenter chacune d'entre elles sous forme d'histogramme de fréquences de mots visuels, en utilisant le même vocabulaire visuel que celui défini lors de la conception de la base. De cette manière il peut ensuite calculer la similarité,  $S_{similarity}$ , entre l'histogramme de la courbe élémentaire et ceux présents dans la base de données. Les modèles 3D possédant des histogrammes avec une similarité élevée, ainsi que l'angle de vue utilisé pour générer le rendu produisant l'histogramme le plus proche, seront sélectionnés par la méthode de récupération comme bons candidats.

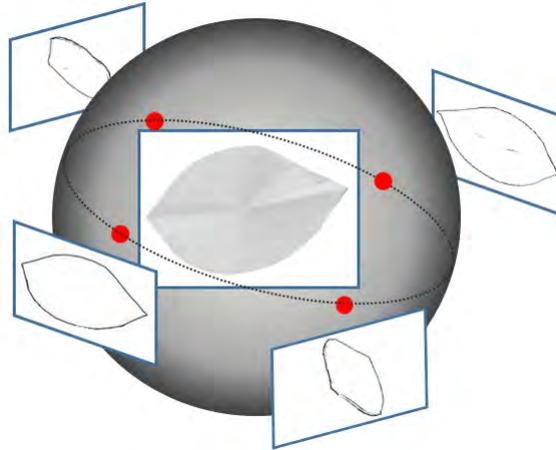


FIGURE 3.6 – Exemple de la génération de 4 rendus d’un pétale 3D selon 4 points de vue placés sur la sphère de vue (point rouge).

Cependant, bien que le simple critère de similarité  $S_{similarity}$  soit suffisant pour des méthodes de récupération travaillant sur des objets dans leur globalité, ce critère n’est pas suffisant pour nous assurer de la fiabilité des modèles 3D récupérés dans notre cas. En effet, puisque nous travaillons sur des éléments botaniques seuls, leurs esquisses peuvent ne pas permettre de déterminer correctement le bon modèle 3D et certains éléments récupérés dans la base pourraient provenir d’une espèce de fleur différente de celle dessinée ou même être d’un tout autre type botanique. La figure fig. 3.7 montre un exemple où l’utilisation de cet unique critère peut entraîner le choix d’éléments d’un mauvais type botanique. En effet l’élément segmenté en (a) peut tout à fait représenter une tige (b), ou une feuille (c), dépendant du dessin dans sa généralité, et la similitude entre cet élément et, par exemple, une tige 3D dans la base de données n’assure pas pour autant que l’élément dessiné représente effectivement une tige.

De manière à nous assurer de bien choisir les éléments 3D qui seront utilisés pour la reconstruction, et donc d’éliminer efficacement les candidats indésirables, nous avons ajusté la méthode de recherche en affectant un score d’évaluation  $S$  défini par :

$$S = S_{similarity} + \alpha \cdot S_{view} + \beta \cdot S_{model} + \gamma \cdot S_{species}$$

où  $\alpha, \beta, \gamma$  sont des constantes.

Comme on peut le constater, notre score d’évaluation part du score de similarité et l’étend en ajoutant plus de critères de sélection. Le premier est le terme  $S_{view}$  qui permet d’accorder une importance à l’angle de vue de l’élément sur le dessin. Au moment de la sélection des modèles 3D dans la base de données, notre système se sert du cône 3D

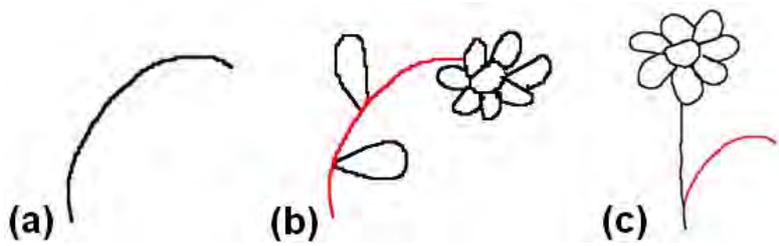


FIGURE 3.7 – Exemple où l’apparence de l’élément botanique seule ne peut permettre de déduire son type. L’élément segmenté en (a) représente dans un cas une tige (b), et dans l’autre une feuille (c). De ce fait, le critère de similarité seule ne suffit pas à s’assurer du bon choix du modèle 3D.

reconstruit (section 3.5) afin d’estimer pour chaque pétale présent dans le croquis son orientation approximative par rapport au point de vue choisi par l’artiste pour le dessin. De cette manière notre système possède maintenant une information supplémentaire pour la courbe de l’utilisateur qu’il peut utiliser pour affiner la recherche de bons candidats. En effet, les rendus 2D des modèles de la base sont générés selon différents angles de projection, et cette donnée est connue de notre système. Par conséquent, lorsque la méthode de récupération basée sur l’esquisse retourne un ensemble de résultats, elle indique également pour chacun d’entre eux le rendu 2D dont la similarité est forte avec le croquis en entrée, et donc l’angle de vue de la projection qui a permis ce rendu. Notre programme va alors chercher à ce que l’orientation estimée du pétale soit aussi proche possible de l’angle de projection du modèle de résultat qui sera sélectionné  $(\varphi, \theta)$ . À cette fin, nous définissons le score de vue  $S_{view}$  par :

$$S_{view} = \exp(-\|\Delta\varphi\|^2/\sigma_\varphi^2) + \exp(-\|\Delta\theta\|^2/\sigma_\theta^2),$$

avec  $\sigma_\varphi = \pi/8$  et  $\sigma_\theta = \pi/4$  en constantes.

Nous avons également ajouté les deux termes  $S_{model}$  et  $S_{species}$  dans le but de favoriser la redondance, respectivement, d’un même modèle 3D et d’un même type de fleur parmi l’ensemble des résultats récupéré depuis la base de données pour tous les éléments botaniques. En effet, les fleurs dessinées par les utilisateurs ont tendance à être composées par des éléments identiques entre eux, notamment pour les pétales, où la majorité du temps l’artiste va dessiner un même type de pétale pour une même couche. La redondance de modèle,  $S_{model}$ , va donc privilégier la sélection d’un élément de la base qui apparaît dans la liste des résultats de multiples courbes élémentaires, c’est-à-dire qu’elle va favoriser l’utilisation d’un même modèle 3D pour représenter des composants différents du modèle final. La redondance d’espèce florale,  $S_{species}$ , est là pour favoriser la sélection d’éléments appartenant à la même espèce de fleur parmi l’ensemble des résultats des courbes élémentaires. Lorsqu’un utilisateur souhaite dessiner une fleur, il a en général une certaine espèce comme modèle en tête, et représente les éléments botaniques en accord avec cette

espèce. C'est pourquoi notre méthode va tenter de privilégier un assemblage d'éléments d'espèce commune. Ainsi, si, par exemple, une majorité d'éléments botaniques composant le modèle final sont détectés comme provenant d'une marguerite, et que parmi les bons candidats pour une feuille se trouve une feuille de rose et une feuille de marguerite, alors le programme aura tendance à favoriser l'utilisation de la feuille de la même espèce que celle des autres éléments, à savoir la feuille de marguerite.

Dans un premier temps nous effectuons une première sélection avec  $\alpha = 1, \beta = 0, \gamma = 0$ . C'est-à-dire que nous commençons par effectuer un premier filtrage basé uniquement sur le critère de similarité et d'angle de vue. Le résultat de ce premier filtrage est ensuite utilisé pour calculer les scores  $S_{model}$  et  $S_{species}$ . Pour ce faire nous comptons le nombre d'occurrences de chaque modèle ou d'espèces de fleurs parmi l'ensemble des résultats, et nous le multiplions par une constante  $c$ , qui a été défini expérimentalement à 0.2 pour  $S_{model}$  et 0.3 pour  $S_{species}$ . Le score d'évaluation de chaque candidat restant est alors mis à jour et réorganisé en mettant  $\alpha = 1, \beta = 1, \gamma = 1$ . De cette manière, un score plus élevé est attribué à un candidat dont le modèle 3D ou l'espèce de fleur est présent à de multiples reprises dans la liste des candidats des différentes courbes élémentaires. Cette stratégie ne va pas pour autant forcer les couches de pétale du modèle finales à être composée uniquement d'un même élément 3D, et il est toujours possible d'obtenir un ensemble de modèles de pétales provenant de différentes espèces de fleurs pour une même couche et laissant ainsi libre cours à la créativité de l'artiste. L'utilisateur dispose cependant d'une option afin de verrouiller l'accès aux pétales d'une couche à une même espèce de fleur s'il désire s'assurer du réalisme de son résultat. S'il décide de l'utiliser, nous sélectionnons alors uniquement les modèles dont l'espèce de fleur a contribué au plus grand nombre de candidats.

Il est également important de préciser qu'au moment de la sélection du modèle parmi la liste des candidats, nous n'avons aucune information sur le type botanique de l'élément représenté par chaque courbe élémentaire, c'est-à-dire s'il s'agit d'un pétale, d'une fleur, d'une tige ou d'un coeur de fleur. De plus la liste des candidats obtenus par la méthode de récupération à l'aide de croquis peut tout à fait être composée d'éléments ayant des types botaniques différents (par exemple, 2 pétales, 1 feuille et 7 tiges), mais dont la projection, selon un certain angle de vue, est pourtant similaire au croquis d'entrée. La sélection du bon candidat pour chaque courbe ne consiste donc pas seulement à prendre le modèle avec le score d'évaluation le plus élevé, il est nécessaire de déterminer le type botanique correct de chaque courbe élémentaire, et de s'assurer de sélectionner un candidat de ce type. Ceci est effectué à l'aide de la méthode par étapes suivante :

1. Si le type botanique dominant dans une liste de candidats est *une tige* et qu'il est la seule courbe élémentaire dont les résultats présentent cette dominance, alors ce type botanique lui est attribué. Ceci se justifie par le fait que notre méthode requiert la présence d'une et une seule tige dans le dessin, et que la forme des tiges se distingue nettement des autres éléments botaniques.

2. Une courbe est considérée comme du type *pétale* si elle comporte au moins un point situé à proximité immédiate de la ou des *courbes d'extrémité* et un autre à proximité du *centre de fleur*.
3. Une courbe dont les candidats contiennent un modèle de type centre de fleur est validée en tant que telle si elle contient le point *centre de fleur* indiqué par l'utilisateur.
4. Les courbes restantes n'étant donc ni pétale, ni centre de fleur, ni tige, se voient attribué le seul type botanique restant traité par notre système, c'est-à-dire qu'elles sont étiquetées comme étant des feuilles.

Une fois que le type botanique de chaque courbe élémentaire est défini, et que les candidats ont été filtrés en conséquence, le modèle avec le score d'évaluation le plus élevé est sélectionné comme étant le bon modèle 3D et servira à composer le modèle final (fig. 3.8).

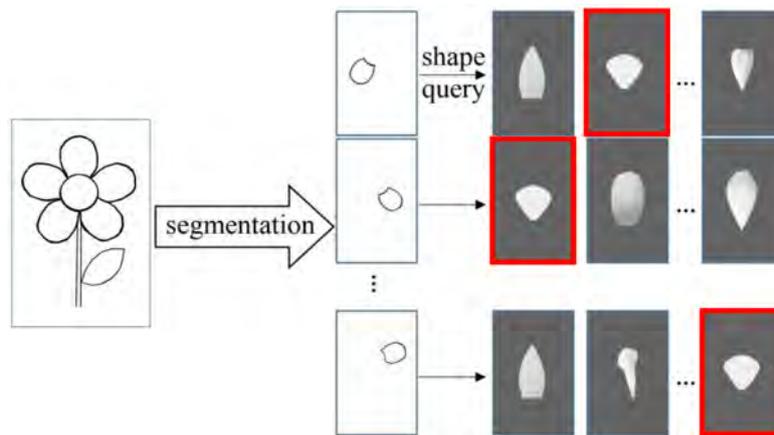


FIGURE 3.8 – Pour chaque courbe représentant un élément botanique, nous récupérons les modèles 3D ordonnés selon leurs scores de similarité et sélectionnons le modèle avec le score d'évaluation le plus élevé combinant la cohérence avec l'angle de vue du dessin, la fréquence de présence d'une même espèce de fleur et d'un même modèle 3D.

### 3.5 Reconstruction du cône 3D

Comme expliqué section 3.1.1 les fleurs partagent une structure commune qui peut être approximée par une forme géométrique simple. En effet chaque couche de pétales d'une fleur peut être modélisée à l'aide d'un cône 3D. Ce sont donc ces cônes que nous cherchons à reconstruire à partir du dessin. Cependant le croquis seul ne permet pas d'avoir les informations nécessaires à cette tâche, c'est pourquoi l'utilisateur doit nous fournir des données supplémentaires par le biais des traits de guidage : le point du *centre*

de fleur  $C_2$ , correspondant au point de jonction entre la tige et la corolle, ainsi qu'une ou plusieurs *courbes d'extrémité*, qui sont les courbes reliant les extrémités des pétales d'une même couche (fig. 3.9). Comme le montre la fig. 3.9 (b), les courbes d'extrémité ne sont pas nécessairement des courbes fermées. En effet, il est demandé à l'utilisateur de ne passer que par les points des pétales visibles sur le dessin, une courbe ouverte indique donc qu'un ou plusieurs pétales de cette couche ne sont pas présents sur l'esquisse, certainement cachés par d'autres éléments tels que la tige ou même par d'autres couches de pétales (comme dans la fig. 3.9 (b)). Afin de pouvoir être utilisée par notre méthode, chaque courbe d'extrémité doit subir une étape de prétraitement qui va chercher à l'approximer par l'ellipse correspondant le mieux [FF70], que l'on nomme *ellipse d'extrémité*. Lorsque la courbe d'extrémité dessinée n'est pas fermée, l'ellipse d'extrémité qui en découle peut être labellisée en 2 parties : la partie visible présente sur le dessin et la partie non visible, qui en est interpolée et permettra par la suite le placement des pétales non présents sur le dessin. Une fois que l'utilisateur a fourni l'ensemble de ces informations, il est possible de reconstruire chaque cône 3D ayant pour base une ellipse d'extrémité et pour apex le centre de fleur.

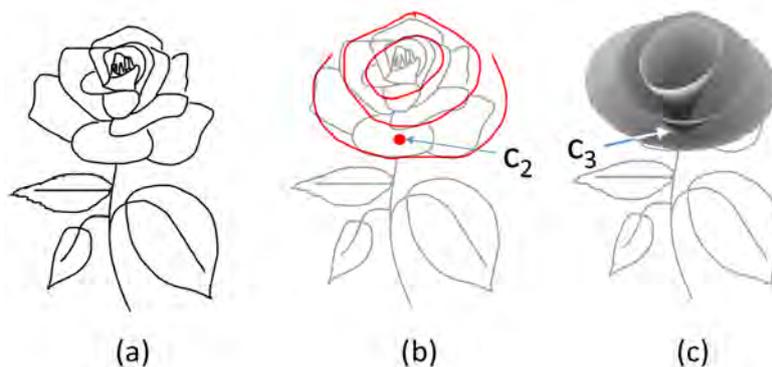


FIGURE 3.9 – Après avoir tracé l'esquisse florale (a), l'utilisateur localise le point *centre de fleur* (éventuellement caché)  $C_2$  où la tige se joint à la corolle (point rouge) et trace les *courbes d'extrémité* (courbes rouges) pour chaque couche de pétales (b). À partir de ces traits de guidage, interprétés comme des ellipses 2D, nous reconstruisons les cônes 3D que nous utiliserons pour guider le positionnement des éléments botaniques 3D (pétales et tige).

### 3.5.1 Orientation de la fleur

La première étape de cette reconstruction est de déterminer l'orientation de la fleur, c'est-à-dire savoir si celle-ci est orientée vers l'observateur ou si elle est de dos par rapport à celui-ci. Cette information est importante puisqu'elle nous indique le sens que les cônes

doivent avoir, c'est-à-dire si ceux-ci doivent être, ou non, orientés vers nous. Cependant, à partir d'une ellipse 2D, correspondant à la projection d'un cercle 3D, et d'un point, ici le centre de fleur qui correspond à la projection de l'apex du cône que l'on cherche, il est possible de reconstruire deux cônes 3D identiques, mais de sens opposés (fig. 3.10). Les traits de guidage seuls ne permettent donc pas de déterminer l'orientation de la fleur, et de résoudre ainsi cette ambiguïté.

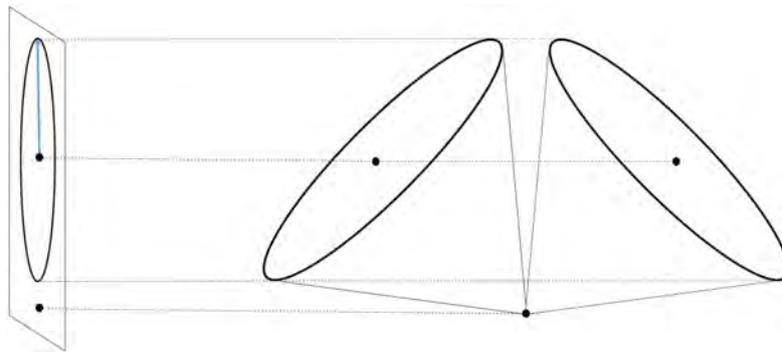


FIGURE 3.10 – Deux cônes 3D identiques, mais de sens opposés possèdent la même projection 2D.

Nous avons alors cherché un moyen de résoudre ce problème le plus simplement possible, en évitant au maximum d'avoir à recourir à de nouvelles interactions avec l'utilisateur et en n'utilisant donc que les informations déjà accessibles au programme, c'est-à-dire le dessin lui-même. Nous avons trouvé que la tige permettait de résoudre ce problème et de lever l'ambiguïté. Lorsque le croquis contient plusieurs couches de pétales, l'utilisateur aura tracé plusieurs courbes d'extrémités, et donc plusieurs ellipses seront disponibles. Nous considérons que la plus grande ellipse est toujours celle que celui-ci aura dessinée le plus précisément possible, et qu'elle transcrit plus précisément l'orientation globale de la fleur. C'est donc cette dernière que nous utilisons.

Définissons  $E_2$  comme étant la plus grande ellipse d'extrémité dont le centre est  $O_2$ , et  $S_2$  le point correspondant à l'extrémité de la courbe de la tige le plus proche de  $O_2$ . Si l'ellipse d'extrémité  $E_2$  est fermée, et donc qu'aucun pétale n'est caché par d'autres éléments, alors la fleur est toujours considérée comme étant orientée vers l'observateur (fig. 3.11 (a)). Dans le cas contraire, nous nous servons de la tige afin de déterminer l'orientation de la fleur. Si le segment  $[S_2, O_2]$  ou la tige elle-même intersecte cette ellipse  $E_2$ , alors nous considérons que les pétales cachés de  $E_2$  le sont par d'autres couches de pétales (tel que c'est le cas fig. 3.9 (b)), et nous en déduisons que la fleur est orientée vers l'observateur (fig. 3.11 (b) et (c)). Sinon cela signifie que les pétales cachés de  $E_2$  le sont par la tige et donc que la fleur est orientée de dos par rapport à l'observateur (fig. 3.11(d)).

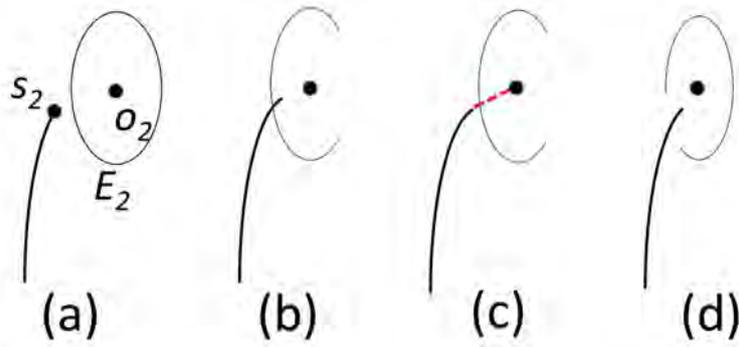


FIGURE 3.11 – La fleur est orientée vers l’observateur si la courbe d’extrémité  $E_2$  est fermée (a) ou si la tige (b) ou le segment  $[S_2, O_2]$  (c) intersecte avec l’une d’entre elles. Sinon, la fleur est orientée de dos par rapport à l’observateur (d).

### 3.5.2 Orientation de la corolle

Maintenant que l’orientation générale de la fleur est déterminée, nous pouvons calculer l’orientation de chaque corolle, qui consiste en l’ensemble des pétales et du coeur de la fleur. Nous supposons que tous les pétales se trouvent sur un cône 3D dont le sommet  $\mathbf{c}_3$  se projette sur  $\mathbf{c}_2$  et dont le cercle de base  $E_3$  se projette sur l’ellipse  $E_2$  sur le plan d’esquisse (fig. 3.12). Nous supposons également que le centre de  $E_3$ ,  $\mathbf{o}_3$  possède sa coordonnée  $z$  fixée à 0. L’axe du cône 3D  $\mathbf{n}_3$  se projette en 2D sur  $\mathbf{n}_2$ , ce qui correspond au petit axe de  $E_2$ . On peut en déduire que  $\mathbf{n}_3 = \mathbf{n}_2 \pm d \cdot \mathbf{z}_3$  où  $d$  est un scalaire. Le deuxième terme pouvant être ajouté ou retiré, puisque deux cônes 3D peuvent être reconstruits à partir d’une même projection, et dépend de l’orientation de la fleur précédemment obtenue (section 3.5.1). Ce terme est ajouté si la fleur fait face à l’observateur et il est soustrait dans le cas contraire. L’angle  $\alpha$  entre  $\mathbf{n}_2$  et  $\mathbf{n}_3$  peut être obtenu par le rapport de  $\mathbf{r}$  et  $\|\mathbf{n}_2\|$ , qui correspondent respectivement à la longueur de l’axe majeur et mineur de l’ellipse. Nous avons donc  $\sin \alpha = \|\mathbf{n}_2\|/\mathbf{r}$  et  $\tan \alpha = d/\|\mathbf{n}_2\|$ . À l’aide des formules trigonométriques on peut alors exprimer  $d$  par rapport à  $\mathbf{r}$  et  $\|\mathbf{n}_2\|$ , qui sont deux informations dont notre système dispose, nous donnant ainsi  $d = \|\mathbf{n}_2\|^2/\sqrt{\mathbf{r}^2 - \|\mathbf{n}_2\|^2}$ .

Connaissant maintenant la valeur de  $d$  nous pouvons calculer la valeur de  $\mathbf{n}_3$ . Cependant, avoir les valeurs de  $\mathbf{o}_3$ ,  $\mathbf{n}_3$  et  $\mathbf{r}$  ne permet de définir que le cercle 3D  $E_3$  qui est la base du cône, il est également nécessaire de calculer par la suite l’emplacement du sommet du cône  $\mathbf{c}_3$ . Ce sommet, ayant pour projection  $\mathbf{c}_2$ , doit être situé sur la ligne  $C_z$  passant par ce point 2D. Puisqu’il est également l’apex du cône 3D il doit aussi se situer sur  $\mathbf{D}_n$ , qui correspond l’axe 3D de la fleur. Par conséquent  $\mathbf{c}_3$  est censé être un point commun à ces deux lignes 3D distinctes, c’est-à-dire qu’il s’agit de leur point d’intersection. Il faut cependant prendre en considération que le dessin de l’utilisateur est très souvent approximatif et ne respecte pas toujours l’intégrité de ces contraintes

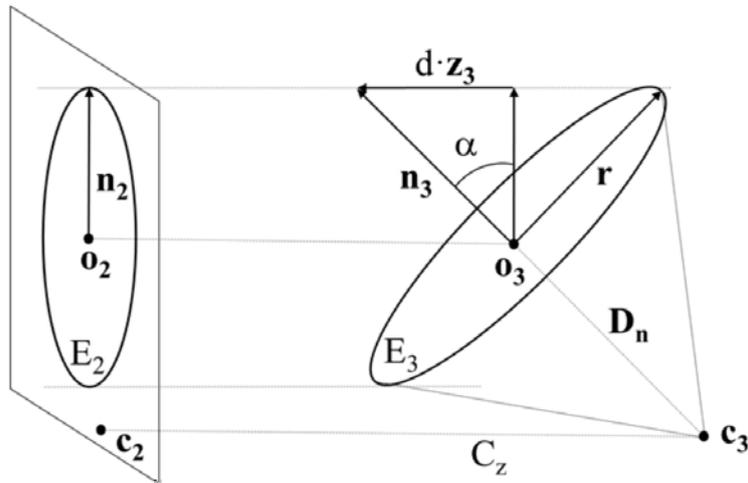


FIGURE 3.12 – Un disque 3D  $E_3$  (la base d'un cône 3D) est projeté sur le plan d'esquisse sous la forme d'une ellipse  $E_2$ .

géométriques. Ces lignes ne se croisent donc pas nécessairement dans la réalité. Dans le cas où aucune intersection n'existe, nous avons décidé de définir  $\mathbf{c}_3$  comme étant le point de  $C_z$  le plus proche de  $\mathbf{D}_n$ . Notons que les cônes obtenus peuvent avoir une ouverture supérieure à 180 degrés, comme celui présent dans fig. 3.1.

Lorsque la fleur présente dans le dessin est composée de multiples couches de pétales, et donc que l'utilisateur a tracé de multiples ellipses d'extrémité, nous devons modéliser plusieurs cônes 3D, un pour chacune des ellipses. L'orientation de chacun des cônes est déterminée de la même manière que lorsqu'il n'y en a qu'un seul, sans tenir compte de la reconstruction des précédentes couches. Par la suite, une translation est appliquée à l'apex de chacun des cônes reconstruits de sorte à le positionner aux coordonnées de celui du premier cône, reconstruit à partir de la plus grande ellipse. Une étape supplémentaire est nécessaire afin de vérifier que les cônes 3D ne s'intersectent pas entre eux. Le cas échéant, le système va réorienter de manière itérative l'axe du cône interne le plus proche des cônes sous-jacents, de manière à éviter les intersections. Ici aussi, la même logique que pour la détection de l'orientation de fleur est appliquée : la plus grande ellipse tracée par l'utilisateur est la plus correcte, c'est donc l'orientation du cône 3D issue de cette ellipse qui restera toujours inchangée en cas d'intersection.

Il existe cependant un cas particulier de dessins qui nécessite un traitement spécifique : les dessins de fleurs d'une vue strictement de face. Dans ce genre de dessin, les ellipses dessinées par l'utilisateur seront des cercles dont le centre se confondra avec le centre de fleur indiqué par l'utilisateur. Ainsi, bien que  $\mathbf{c}_3$  et  $\mathbf{o}_3$  soient deux points distincts dans l'espace 3D, ils se projettent au même point 2D (ou en deux points très proches)

rendant impossible le calcul de leur coordonnée  $z$ , et donc de la profondeur du cône. Nous avons donc décidé que lorsque la différence entre  $\mathbf{r}$  et  $\mathbf{n}_2$  est négligeable, c'est-à-dire que  $|\mathbf{r} - \mathbf{n}_2| < 0.1 \cdot \mathbf{r}$ , alors les informations de profondeur extraites des ellipses ne sont pas fiables pour calculer automatiquement le cône 3D et choisissons arbitrairement un angle d'ouverture de 140 degrés pour le premier cône. Si la fleur dessinée de face est composée de multiples couches de pétales alors nous supposons également que les plus petits cônes sont plus proches de l'observateur que les plus grands, et plaçons donc le centre de leur base plus proche de nous.

## 3.6 Création du modèle 3D par assemblage des éléments botaniques

La prochaine étape est de reconstruire la fleur désirée en plaçant les éléments floraux préalablement récupérés autour du (ou des) cône(s) 3D obtenu(s) à la section 3.5. Les éléments correspondant aux pétales et au centre de la fleur, le maillage 3D faisant office de réceptacle simplifié, sont directement positionnés à l'aide du cône 3D et de leurs esquisses (section 3.6.1). Les autres éléments floraux, à savoir la tige et les feuilles, sont placés par la suite (section 3.6.2). Une fois que tous les éléments sont positionnés, notre système procède à une dernière étape de raffinement qui consiste à déformer le maillage des pétales de manière à ce que ceux-ci correspondent parfaitement au dessin réalisé par l'utilisateur (section 3.6.3), permettant ainsi d'avoir un modèle 3D final dont la projection correspond parfaitement à l'esquisse.

### 3.6.1 Placement des pétales

Nous devons déterminer la position dans l'espace 3D de chaque courbe élémentaire 2D dont le type botanique a été précédemment déterminé comme étant un pétale. Nous avons défini que chaque couche de pétale peut être approximée par un cône 3D, que nous avons reconstruit grâce aux traits de guidage, les ellipses d'extrémités, fournies par l'utilisateur. Nous avons donc une correspondance établie entre les ellipses 2D et leur équivalent en 3D, c'est-à-dire les bases des cônes. Par conséquent, déterminer la position d'un pétale revient à chercher la position de sa pointe sur la courbe d'extrémité 2D, en cherchant simplement les coordonnées 2D (x,y) du point de l'ellipse d'extrémité le plus proche de la pointe du pétale, et en cherchant le point du cercle 3D correspondant à cette ellipse. On obtient ainsi directement la position 3D sur la base du cône à laquelle la pointe du pétale devra être placée.

Maintenant que la position de la pointe 3D du pétale est définie, nous devons déterminer son orientation par rapport au cône. Puisque dans une fleur, les pétales poussent radialement à partir du centre, nous devons déterminer deux angles d'orientation pour chaque pétale (fig. 3.13) : l'angle de floraison et l'angle de roulis. L'angle de floraison d'un pétale correspond à l'angle entre l'axe de la fleur et l'axe défini par les 2 extrémités du pétale (celle vers le centre de la fleur et celle sur la base du cône 3D), plus il est

grand et plus la fleur est ouverte. Cet angle est en réalité directement donné par le cône 3D, et est déterminé de manière à ce que la ligne reliant les 2 extrémités du pétale soit située sur sa surface. Puisque la position 3D d'une des deux extrémités du pétale est déjà déterminée, la pointe sur la base du cône 3D, l'angle de floraison permet maintenant de placer la seconde extrémité du pétale. Il reste alors à définir l'angle de roulis, correspondant à l'orientation autour de l'axe longitudinal du pétale. Ce dernier est déterminé de sorte que le vecteur normal à la base du pétale intersecte l'axe du cône. Ce choix a été fait pour plusieurs raisons. L'angle de roulis est nécessaire afin de s'assurer que le pétale soit bien orienté par rapport à l'orientation générale de la fleur, et à ce que le résultat final, c'est-à-dire le modèle 3D composé de l'ensemble des éléments floraux, soit réaliste. Le choix de prendre le vecteur normal à la base du pétale, et pas ailleurs, a été fait du à l'existence de pétale avec des torsions, faisant de la base du pétale la seule zone commune à toutes les espèces botaniques, et qui sera toujours orientée en direction du centre de la fleur.

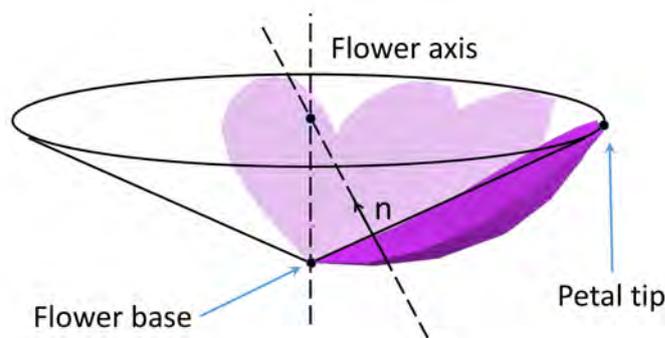


FIGURE 3.13 – Placement des pétales autour d'un cône 3D.

Une fois que tous les pétales sont placés, la corolle est complétée par l'ajout du modèle 3D correspondant au centre de la fleur, si un tel élément botanique est présent dans l'esquisse. Le modèle 3D correspondant au centre de la fleur est placé au centre des pétales 3D, au niveau de l'apex du cône. Les pétales subissent par la suite un léger réajustement de leur placement 3D afin de bien s'adapter au centre de la fleur. Plus précisément, nous trouvons la translation de chaque pétale, le long de sa direction longitudinale, de sorte que le centre de la fleur et la base du pétale soient en contact direct.

**Gestion des pétales cachés.** La méthode de positionnement susmentionné décrit comment est effectué le positionnement des pétales dans les cas généraux, mais n'explique pas comment sont gérés les dessins avec des pétales cachés. Lorsque la corolle représentée dans l'esquisse n'est pas complète, c'est-à-dire que certains pétales ne sont pas présents

en raison d'une occultation due à d'autres éléments botaniques (tel que dans fig. 3.1), l'ellipse d'extrémité est dessinée ouverte et notre système doit alors déterminer deux informations non présentes directement dans l'esquisse : le nombre de pétales manquants, et leurs positions. La stratégie employée pour déterminer le nombre approximatif de pétales manquants, ainsi que leur positionnement 3D, se base sur le constat que la distance entre les pétales d'une même couche de fleur est quasiment constante, et que calculer cette distance nous permettra d'en déduire le nombre de pétales manquant et de les placer. Notre système procède donc de la manière suivante :

1. Nous commençons par détecter les extrémités des pétales visibles et les localisons sur l'ellipse d'extrémité 2D, obtenant par la même façon leurs positions sur la base du cône 3D correspondant.
2. Nous mesurons la distance moyenne entre les pétales voisins sur la circonférence de la base du cône 3D.
3. Nous mesurons la longueur de l'ellipse d'extrémité labellisée comme non visible, et nous divisons cette longueur par la distance moyenne entre les pétales visibles. De cette manière, nous obtenons le nombre de pétales manquants qui doivent être ajoutés.
4. À partir du dernier pétale visible, nous plaçons des repères sur la partie de l'ellipse non visible correspondant aux emplacements des pétales manquants. Ces repères sont placés en respectant la distance moyenne des pétales visibles.

Une fois cette étape effectuée, notre modelleur sait combien de pétales 3D il doit placer, et où les positionner. Le modèle 3D de pétale le plus présent parmi ceux utilisés pour représenter les pétales dans la partie visible de cette corolle est alors utilisé par défaut comme modèle pour les pétales non visibles du dessin. Grâce à cette méthode, notre système est capable de reconstruire une fleur complète, malgré l'absence de pétales sur le croquis due aux occultations.

### 3.6.2 Positionnement de la tige et des feuilles

Une fois que l'ensemble des éléments composant la partie florale, c'est à dire les pétales ainsi que le centre de fleur, sont bien positionnés, il ne reste plus qu'à déterminer l'emplacement de la tige et des feuilles. Notre système commence par positionner le modèle 3D de la tige précédemment récupéré de la base de données. Ceci est trivial et se fait, comme pour les pétales, directement à l'aide du cône 3D. En effet l'apex du cône 3D correspond au point de *centre de fleur* indiqué par l'utilisateur, et qui a été défini comme étant le point de jonction entre la corolle et la tige. Les coordonnées 3D de ce point sont déjà connues, puisque le cône 3D est déjà reconstruit. La tige est donc déplacée de manière à ce que sa première extrémité se situe au niveau de l'apex du cône 3D et à ce que sa seconde extrémité possède les mêmes coordonnées x,y que le bas de la tige sur le dessin (puisque nous avons une projection orthogonale, c'est-à-dire que les éléments 2D ont les mêmes coordonnées x,y que leurs homologues 3D).

Le placement des feuilles est également une étape facile. Ici, l'orientation de la feuille est directement récupérée dans la base de données, et correspond à l'angle de vue utilisée pour produire le rendu ayant donné le meilleur score de similarité. Il ne reste donc qu'à placer le modèle 3D de la feuille aux bonnes coordonnées. Puisque le dessin est une projection orthogonale de la fleur 3D, il suffit de regarder les coordonnées  $x,y$  des points de jonction entre la tige et les feuilles sur celui-ci, puis de chercher sur la tige 3D précédemment placée les points appartenant à sa zone d'attachement avec les coordonnées  $x,y$  les plus proches possibles. C'est à ces points-là que l'on attachera les modèles 3D des feuilles à la tige. Une fois attaché, il ne reste plus qu'à placer le haut de la feuille de manière à ce que sa pointe se situe le plus proche possible des coordonnées 2D qu'elle a sur le dessin.

### 3.6.3 Raffinement du maillage des pétales

Dans la plupart des cas, la fleur dessinée par l'utilisateur est composée d'un ensemble de pétales identiques, c'est-à-dire issus d'une même espèce. Cependant, bien qu'il s'agisse de pétales identiques, ils peuvent tout de même différer les uns des autres, sur le dessin, par une forme ou une flexion légèrement différente et propre à chacun d'entre eux. À contrario, le modèle 3D récupéré dans la base de données pour représenter ces pétales est identique pour chacun d'entre eux (puisque nous privilégions la redondance d'un même modèle 3D pour une même couche de pétale, comme expliqué section 3.4.2), par conséquent ces petites différences entre les pétales, pourtant bien visibles sur le dessin, ne seront pas présentes dans le résultat final. De plus, puisque notre base de données contient un nombre limité d'éléments floraux, il n'est pas toujours possible de trouver un élément 3D qui correspond parfaitement à l'esquisse de l'utilisateur, et le modèle retourné sera uniquement *le plus proche possible* de ce que voulait l'utilisateur, entraînant ainsi une différence notable entre le modèle final reconstruit et le dessin.

Afin de faire face à ces deux cas, nous appliquons une étape de raffinement du maillage des pétales. Au cours de cette étape, une déformation non rigide est appliquée à chaque pétale 3D de sorte qu'il corresponde parfaitement à son vis-à-vis 2D. De cette manière, le modèle de fleur final est totalement fidèle au dessin 2D, et chaque pétale 3D se trouve alors légèrement différent des autres en présentant les mêmes caractéristiques (forme et flexion) que son homologue 2D. Afin d'effectuer les déformations, nous procédons en suivant les étapes suivantes.

**Établissement d'une correspondance point à point entre le dessin 2D et le modèle 3D.** Afin d'effectuer la déformation non rigide du pétale, nous devons définir la nouvelle position que doit avoir chaque point 3D du maillage afin d'obtenir la forme désirée. Ceci est effectué en établissant un lien de correspondance point à point entre la silhouette 2D du pétale dessiné par l'utilisateur et la silhouette du modèle 3D utilisé pour la reconstruction. Dans un premier temps, définissons  $D$  comme étant le contour du pétale dessiné par l'utilisateur (fig. 3.14 a). Par la suite, cherchons  $P_t^D$  et  $P_b^D$ , respec-

tivement la pointe et la base du pétale. Ces deux points sont obtenus en utilisant la courbe d'extrémité et le centre de fleur fourni par l'utilisateur. La pointe du pétale,  $P_t^D$ , correspond au point de  $D$  étant à la fois le plus proche de l'ellipse de l'extrémité et le plus éloigné du centre de fleur, puisqu'il s'agit de l'extrémité du pétale. Quand au point  $P_b^D$  il correspond au point de  $D$  le plus proche du centre de fleur, puisqu'il s'agit du point de base. Définitions ensuite  $S$  comme étant la projection de la silhouette  $S_{3D}$  du modèle 3D du pétale correspondant sur le plan d'esquisse ( $z=0$ ). Les points caractéristiques de  $S$ ,  $P_t^S$  et  $P_b^S$  sont facilement trouvables puisqu'il s'agit des deux points d'extrémités du pétale 3D qui sont listés dans le fichier de métadonnées associé au modèle 3D dans la base. Nous pouvons maintenant établir une correspondance point à point entre  $S$  et  $D$ . Tout d'abord la correspondance est établie entre les points de base et les points de pointes, c'est-à-dire entre  $(P_t^D, P_t^S)$  et  $(P_b^D, P_b^S)$ . Il est alors possible de séparer  $D$  en deux segments de courbes  $[P_t^D P_b^D]$  et  $[P_b^D P_t^D]$ . La même segmentation peut être effectuée pour  $S$ . De cette façon nous avons maintenant deux segments pour chaque silhouette avec une correspondance entre leurs extrémités (et donc une correspondance entre les segments eux-mêmes). Chacun de ces segments de courbe est alors échantillonné uniformément de sorte que chaque segment en correspondance contienne le même nombre de points. On peut ensuite établir aisément une correspondance entre les points échantillonnés de  $S$  et ceux de  $D$  : en partant d'une des extrémités d'un segment de courbe, le correspondant d'un point  $p_{S,i}$  de  $S$  est le point  $p_{D,i}$  du segment correspondant de  $D$  et dont l'ordre de position sur ce segment est le même que celui du point de  $S$  (par exemple le 3e point du segment  $[P_b^S P_t^S]$  sera mis en correspondance avec le 3e point du segment  $[P_b^D P_t^D]$ ).

Puisque nous avons comme donnée d'entrée un croquis d'une fleur de n'importe quelle espèce et complexité, et selon n'importe quel angle de vue, il est possible qu'il contienne des pétales qui ne soient pas dessinés intégralement. C'est le cas par exemple lorsque la fleur contient de multiples couches de pétales, et que la couche inférieure est en partie cachée par la couche supérieure (fig. 3.14 d). Dans de tels cas, la correspondance établie ne pourra être faite que partiellement, et approximativement. On détectera cependant toujours un point caractéristique correspondant à la pointe du pétale  $P_t^D$  (puisque s'il n'y a pas de point de pétale, cela signifie qu'on ne voit pas le pétale du tout), et on commencera par établir une correspondance entre celle-ci et son équivalent sur la silhouette du modèle 3D  $P_t^S$ . On connaîtra également toujours la position indiquée par l'utilisateur comme étant le centre de la fleur, et on peut s'en servir comme  $P_t^B$  approximatif. À partir de là, il est alors possible de joindre la partie visible du pétale à  $P_t^B$ , proposant ainsi un contour complet temporaire qui permettra d'effectuer l'étape d'échantillonnage uniforme nécessaire à la mise en correspondance point à point entre le dessin et le modèle 3D, comme expliqué précédemment. Cependant, ici, une fois cet échantillonnage effectué, seuls les points sur la partie visible du pétale dessiné sont mis en correspondance avec leur vis-à-vis sur le modèle 3D, les autres points du modèle 3D, eux, seront alors considérés comme n'ayant pas de position (x,y) imposée par le dessin, et leur position sera automatiquement déterminée par la déformation.

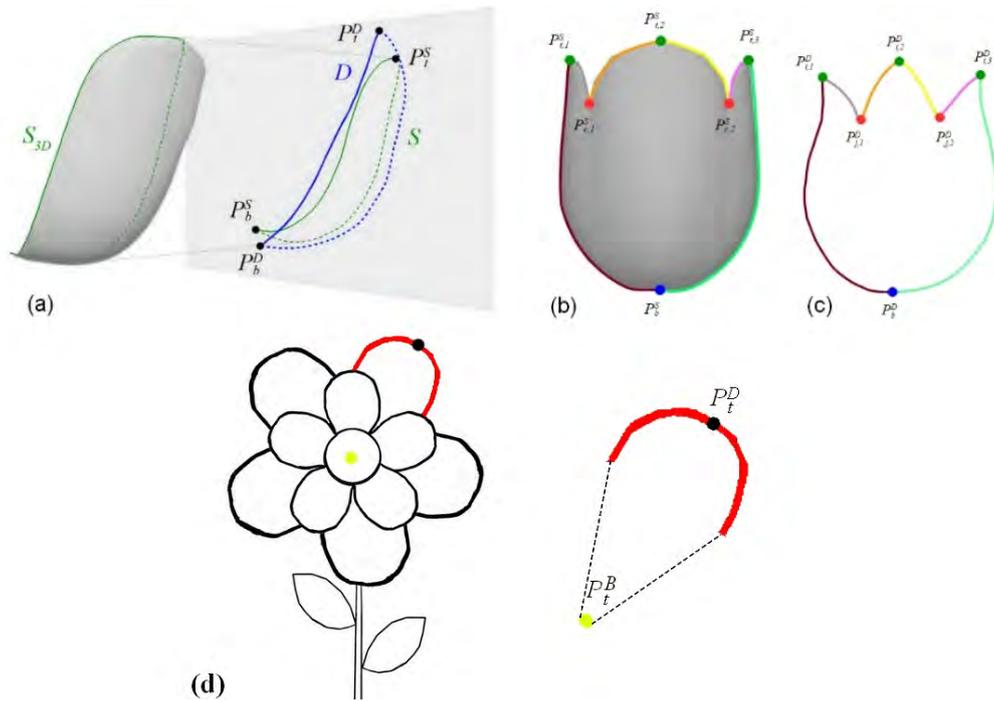


FIGURE 3.14 – (a) montre la mise en correspondance entre  $S_{3D}$ , la silhouette 3D du pétale, et  $D$ , la courbe dessinée à la main du pétale par l'utilisateur, en se servant de  $S$ , la projection sur le plan d'esquisse de  $S_{3D}$ .  $P_t^S, P_b^S, P_t^D$  et  $P_b^D$  sont les points caractéristiques de  $S$  et  $D$  respectivement. Les deux segments indiqués en pointillés et les lignes continues sont échantillonnés de manière à être composés du même nombre de sommets. (b) et (c) montre un processus similaire dans le cas où la corolle complète du dessin n'est représentée que par un seul trait. Lorsque le dessin de pétale n'est pas complet, principalement car il est caché par d'autres éléments de la fleur, seule la partie visible du pétale est prise pour établir la correspondance en utilisant le seul point caractéristique que l'on a (d).

Il est également important de rappeler que l'utilisateur dispose d'une certaine liberté dans la façon de dessiner la fleur, et peut ainsi réaliser une esquisse avec un niveau d'abstraction plus élevé en représentant, par exemple, plusieurs pétales à l'aide d'une seule courbe (fig. 3.14 b et c). Dans ce cas on ne peut effectuer une correspondance directe entre la courbe  $D$  et la silhouette du pétale 3D  $S$  car  $D$  ne va pas représenter ici qu'un seul pétale. Une différente stratégie est donc utilisée afin d'établir la correspondance. Redéfinitions  $D$  pour ce cas précis comme étant maintenant la courbe du dessin représentant cette fois-ci plusieurs pétales. Plusieurs points caractéristiques doivent alors être identifiés dans  $D$ . Certains d'entre eux sont obtenus directement par l'utilisateur. En effet  $P_b^D$ , le point de base de la fleur correspond au point de centre de fleur fournit par celui-ci, de plus les pointes des  $n$  pétales composants  $D$ ,  $P_{t,i}^D$  avec ( $i=1, \dots, n$ ), sont

également fournis par l'utilisateur lors de la segmentation de cette courbe représentant plusieurs pétales (section 3.4.1). Ainsi les seuls points caractéristiques manquants sont les *points de coin*  $P_{c,j}^D$  avec  $(j=1,\dots,n-1)$ , c'est-à-dire les points qui marquent le passage de la silhouette d'un pétale à la silhouette d'un autre. Ceux-ci ont été détectés automatiquement en utilisant la méthode de détection d'arêtes et de points coins présentée par Chris Harris ([HS88]).

Nous redéfinissons aussi  $S_{3D}$  et  $S$  pour ce cas particulier comme étant cette fois-ci non plus la silhouette 3D, et sa projection 2D, d'un pétale, mais comme celle de l'ensemble des pétales 3D représenté par  $D$ . Tout comme dans le cas normal, les points caractéristiques correspondant à la base et à la pointe de chacun de ces pétales 3D est obtenu via leurs métadonnées, et directement retrouvable sur  $S$  par simple projection. Les points de coins  $P_{c,j}^S$  avec  $(j=1,\dots,n-1)$  sont ici trouvés en calculant l'intersection entre les courbes de la silhouette provenant de deux pétales voisins. À ce moment nous disposons donc de deux ensembles de points caractéristiques, un pour  $D$  et un pour  $S$ , et nous établissons une correspondance entre ces points caractéristiques en utilisant une méthode de préservation de l'ordre des points caractéristiques ([SN06]). Par la suite on peut procéder de la même façon que dans le cas simple, en échantillonnant uniformément chaque segment de courbe (délimité par deux points caractéristiques successifs) on peut établir une correspondance point à point entre  $S$  et  $D$

**Déformation laplacienne.** Maintenant que nous avons calculé la correspondance point à point entre la silhouette du maillage de pétale et la courbe de l'esquisse correspondante, nous pouvons déformer le modèle 3D de manière à déplacer les points de la silhouette aux emplacements cibles correspondants sur le dessin de l'utilisateur. Plusieurs types de déformation de maillage 3D existent. On pourrait penser par exemple à utiliser une déformation spécifique à une classe de modèle 3D *pétale* en utilisant une méthode telle que [PFZVG10]. Cependant bien que nous ne déformions qu'un seul type d'élément botanique, les pétales, la forme de cet élément dépend de l'espèce de la fleur d'origine, et une seule classe ne pourrait représenter, à elle seule, l'ensemble des pétales des différentes espèces. Par conséquent, cette méthode demanderait une étape de pré-traitement considérable, car une classe déformable pour les pétales de chaque espèce devrait être précalculée.

Nous avons choisi d'utiliser une déformation laplacienne générique ([SCOL<sup>+</sup>04]) pour sa capacité à préserver les détails du maillage au cours de la déformation. Cette caractéristique nous semblait essentielle pour permettre à notre résultat d'être le plus réaliste possible. En effet, notre méthode se base sur une base de données d'éléments floraux permettant de récupérer des modèles 3D de pétales de différentes espèces de fleurs, et présentant des caractéristiques propres à cette espèce (forme, pliage du pétale, rugosité du maillage, etc.) que nous désirons retrouver sur le modèle final, après déformation. Il était donc nécessaire que la déformation conserve l'ensemble des détails locaux du

maillage, ce que fait la déformation laplacienne. De plus le calcul de cette opération est linéaire, et donc peu coûteux.

Les coordonnées  $(x, y)$  d'un point de la courbe de silhouette  $S_{3D}$  sont alignées sur les coordonnées de son homologue correspondant sur la courbe  $D$ , tandis que sa coordonnée  $z$  reste inchangée. Ces points de silhouette ainsi placés à des coordonnées fixes servent alors de contraintes à la déformation laplacienne appliquée à l'ensemble du modèle 3D du pétale (fig. 3.15). Dans le cas où l'esquisse du pétale ne serait pas complète, lorsqu'une partie de celui-ci est cachée par un autre élément par exemple, seuls les points visibles du dessin servent de contraintes à la déformation, et la partie du maillage 3D sans correspondance 2D ne sera que très légèrement modifiée, restant le plus proche possible de la forme initiale du pétale.

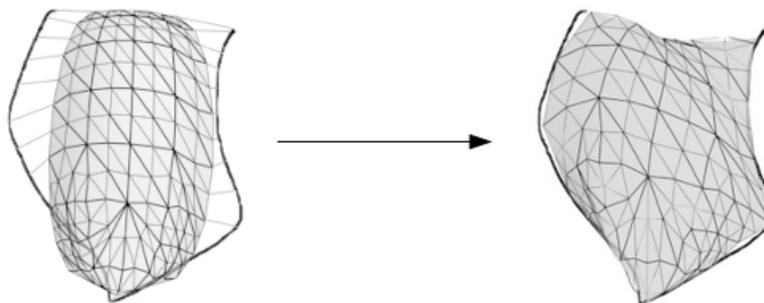


FIGURE 3.15 – Pour chaque point de la silhouette du maillage 3D du pétale, une correspondance est établie avec un des points de la courbe du dessin (a). Grâce à cette correspondance, une déformation de laplacien est appliquée, permettant de raffiner le modèle pour qu'il corresponde parfaitement à l'attente de l'utilisateur (b).

### 3.7 Résultats

Notre modéleur a été implémenté en C++ en tant qu'une application s'exécutant sur un ordinateur avec Windows 10 comme système d'exploitation. Tous nos tests ont été effectués sur une machine dotée de 16 Go de mémoire, avec un processeur Intel Core i7-6700 cadencé à 3,40 GHz. L'ordinateur est équipé d'une tablette-écran tactile Cintiq-13HD-S02 de la marque Wacom. C'est sur cet écran que l'utilisateur effectuera le croquis à l'aide d'un stylo numérique.

La figure 3.16 montre un aperçu de l'interface utilisateur de notre système, qui se compose de deux parties. La première fenêtre, située à gauche, consiste en la zone 2D. Il s'agit de la partie dans laquelle l'utilisateur est invité à dessiner la fleur qu'il désire reconstruire et dans laquelle il fournit les traits de guidage. Il est également possible

pour l'utilisateur d'importer un dessin vectoriel préexistant dans cette fenêtre plutôt que de dessiner, bien que, dans ce cas-là, nous ne disposerions pas des informations temporelles sur la façon dont a été réalisé le dessin, et la segmentation automatique pourrait échouer, nécessitant que l'utilisateur dessine des modèles référents pour les éléments floraux mal segmentés (section 3.4.1). La deuxième partie, située à droite, consiste en la zone 3D. C'est dans cette fenêtre que le résultat final 3D sera montré à l'utilisateur. Il aura la possibilité de tourner le modèle 3D pour le voir selon n'importe quel angle, et de zoomer/dézoomer comme dans n'importe quelle autre application 3D. Une option permet également à celui-ci d'afficher ou cacher les cônes 3D qui ont été reconstruits à l'aide des ellipses d'extrémité.

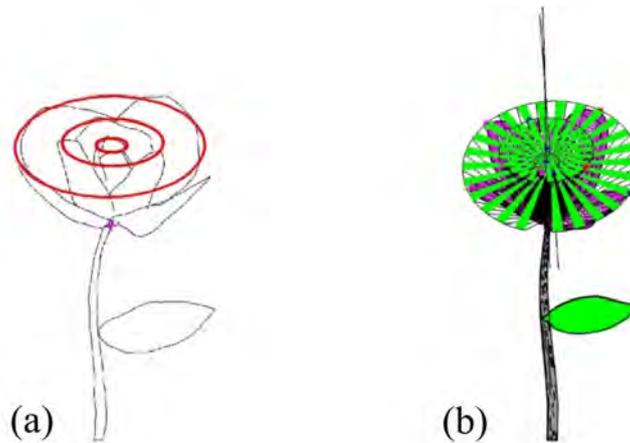


FIGURE 3.16 – L'interface utilisateur est composée de deux fenêtres. La première montre l'esquisse de l'utilisateur avec le centre de fleur (la croix rose) indiquée par ce dernier. On peut également y voir les trois ellipses calculées à partir des courbes d'extrémités fournies par l'utilisateur (a). La deuxième fenêtre (b) montre le résultat de la reconstruction, le modèle final 3D, dans un rendu en fil de fer, et propose également une visualisation des cônes 3D déduits à partir des ellipses (la base des cônes étant les cercles partiellement remplis en vert).

Afin de valider l'efficacité de notre système, nous l'avons testé à l'aide d'une variété de croquis représentant des fleurs provenant de multiples espèces, et représenté selon différents points de vue. Les dessins et les traits de guidage pour leur reconstruction ont été fournis par de multiples utilisateurs âgés de 10 à 46 ans avec des compétences en dessin dans la norme (aucun artiste). Certains résultats obtenus avec notre modèleur sont présents fig. 3.17, et d'autres sont présentés dans l'annexe correspondante (appendix A). Comme on peut le voir, que les fleurs possèdent une seule couche de pétales (fleur à 5 pétales, fleur à pétales...) ou de multiples couches (fleur à deux couches, fleur aux pétales tombants, Jasmin lionhead,...) notre système les reconstruit avec succès. On

peut également constater que leur reconstruction est correcte, peu importe le point de vue choisi par l'artiste : la fleur à 5 pétales est vue de face, la fleur à 9 pétales de derrière et la tulipe ou la fleur aux pétales tombants sont vues de profil. Ces deux derniers modèles permettent également de mettre en avant que les différents degrés de floraison possibles sont bien gérés par notre système, la tulipe étant presque pleinement fermée sur elle-même et la fleur aux pétales tombants étant, à contrario, plus qu'ouverte avec certains de ces pétales pointant vers le bas de la fleur. De manière remarquable, notre modeleur est capable de traiter des fleurs plus sophistiquées, possédant un grand nombre de couches de pétales (Jasmin Lionhead), ou avec des pétales ayant des tailles variables (fleur imaginaire). D'autres modèles présents dans l'annexe montrent également des fleurs avec des pétales aux formes complexes ou se repliant sur eux-mêmes. Enfin, malgré le fait que le croquis de la tulipe soit à un niveau d'abstraction assez élevé, notre programme parvient à proposer une bonne reconstruction de la fleur, bien que du au quatrième pétale, non présent dans le dessin et ajouté automatiquement lors de la reconstruction, la projection du résultat final donne une silhouette légèrement différente au dessin 2D de l'artiste.

La seconde colonne de la fig. 3.17 montre le résultat reconstruit à partir des éléments floraux 3D récupérés directement depuis la base de données, sans passer par l'étape de la déformation. On peut voir que dans certains cas le modèle de pétale utilisé est vraiment très proche de ce que l'utilisateur a dessiné (Fleur à 9 pétales, Tulipe, Fleur à deux couches...). Pour d'autres exemples, on constate une légère différence entre le dessin et le résultat 3D (fleur à 5 pétales, fleur imaginaire, fleur aux pétales tombants). Dans ces cas, la différence est due au nombre limité d'éléments floraux présents dans la base, limitation de notre méthode qui est corrigée par l'étape de déformation. Cependant, l'importance de cette étape est vraiment mise en avant pour des fleurs complexes telles que le Jasmin Lionhead. On remarque que le modèle de pétale sélectionné varie énormément de ceux présents dans le croquis. Ceci est dû à la présence d'un grand nombre de pétales partiellement cachés. Comme expliqué section 3.4.1, la méthode de récupération depuis la base de données à partir d'un croquis va rechercher un modèle 3D pour chaque pétale, y compris ceux dont une partie est cachée en ne recherchant donc qu'une correspondance partielle. Par conséquent, lorsque l'esquisse contient de nombreux pétales ne montrant que leurs extrémités hautes, un modèle de pétale 3D correspondant le mieux à ces pointes sera privilégié comme étant le modèle à utiliser pour la reconstruction, même si sa forme est différente de celle des pétales complètement présents dans le dessin. Cependant, la déformation corrige pleinement ce problème.

Le temps nécessaire pour réaliser les esquisses pour ces résultats dépend du niveau de compétence en dessin de l'utilisateur, mais également de la complexité de la fleur représentée. Il varie donc entre quelques secondes pour les plus simples à cinq minutes pour le plus complexe (Jasmin Lionhead). Une fois l'esquisse fournie il n'a fallu que quelques dizaines de secondes d'interactions avec l'utilisateur pour obtenir les informations de guidage (*courbes d'extrémité, centre de fleur*) nécessaire à la reconstruction, y

compris pour le modèle le plus complexe. Toutes les étapes ultérieures, c'est-à-dire la segmentation de l'esquisse en éléments botaniques, la récupération des éléments floraux 3D depuis la base de données, leur placement sur les cônes 3D estimés et la déformation pour correspondre parfaitement au dessin se font dans des temps relativement faibles. Bien que ce ne soit pas instantané, ces étapes ne demandent que quelques secondes, y compris pour les modèles complexes (30 secondes pour la Jasmin Lionhead qui est composée de 44 éléments floraux). Précisons cependant qu'un temps d'interaction plus élevé est requis pour les dessins qui ne sont pas segmentables automatiquement (comme la tulipe), puisque l'utilisateur doit passer par une étape supplémentaire où il est invité à dessiner le croquis d'un pétale référent puis guider la segmentation via quelques clics.

### 3.7.1 Limitations

Notre méthode se sert d'une base de données contenant divers éléments floraux 3D pour permettre la reconstruction du modèle final. C'est ce qui fait l'une des forces de notre système et permet de s'assurer d'avoir des résultats de qualité, et réalistes. Cependant, c'est également une des limites de notre modèleur, l'expression géométrique du modèle reconstruit se retrouve restreinte au contenu de la base. Bien que la déformation des pétales réduise considérablement ce problème, si l'utilisateur venait à dessiner des pétales d'une forme totalement différente de n'importe quel élément 3D présent dans la base, l'étape de récupération d'un modèle 3D pourrait complètement échouer et ne retourner aucun maillage de pétale. La reconstruction serait alors impossible.

Une autre limitation de notre technique, inhérente aux méthodes de reconstruction à partir d'une vue unique, est révélée lorsque des pétales qui devraient être convexes sont esquissés d'une vue strictement frontale (vue pleinement de profil par exemple fig. 3.18). Dans ce cas, les courbes obtenues par la segmentation du dessin en éléments botaniques ne représentent pas bien la courbure 3D que devraient avoir les pétales. La récupération basée sur l'esquisse renverra des modèles de pétales dont la projection 2D maximisera la similarité avec l'esquisse, mais comme l'esquisse, dû à la vue frontale, ne décrit pas la courbure 3D qu'ils devraient avoir, les pétales retrouvés peuvent être plutôt plats et ne pas correspondre totalement à l'intention du dessinateur. Heureusement, les utilisateurs ont tendance à choisir des vues plus *informatives* dans un tel cas, c'est-à-dire des vues qui mettent en avant les caractéristiques particulières de la fleur : des vues non frontales révélant naturellement la courbure que les pétales doivent avoir. Enfin, ce qui peut être vu comme une limitation minimale, la présence d'une tige est nécessaire dans l'esquisse de la fleur, puisque nous utilisons sa position relative avec la corolle afin de déterminer l'orientation du cône 3D.

## 3.8 Conclusion

Ce chapitre a présenté la première solution mise au point au cours de ma thèse pour la modélisation géométrique par croquis en mettant au point un système dédié à la re-

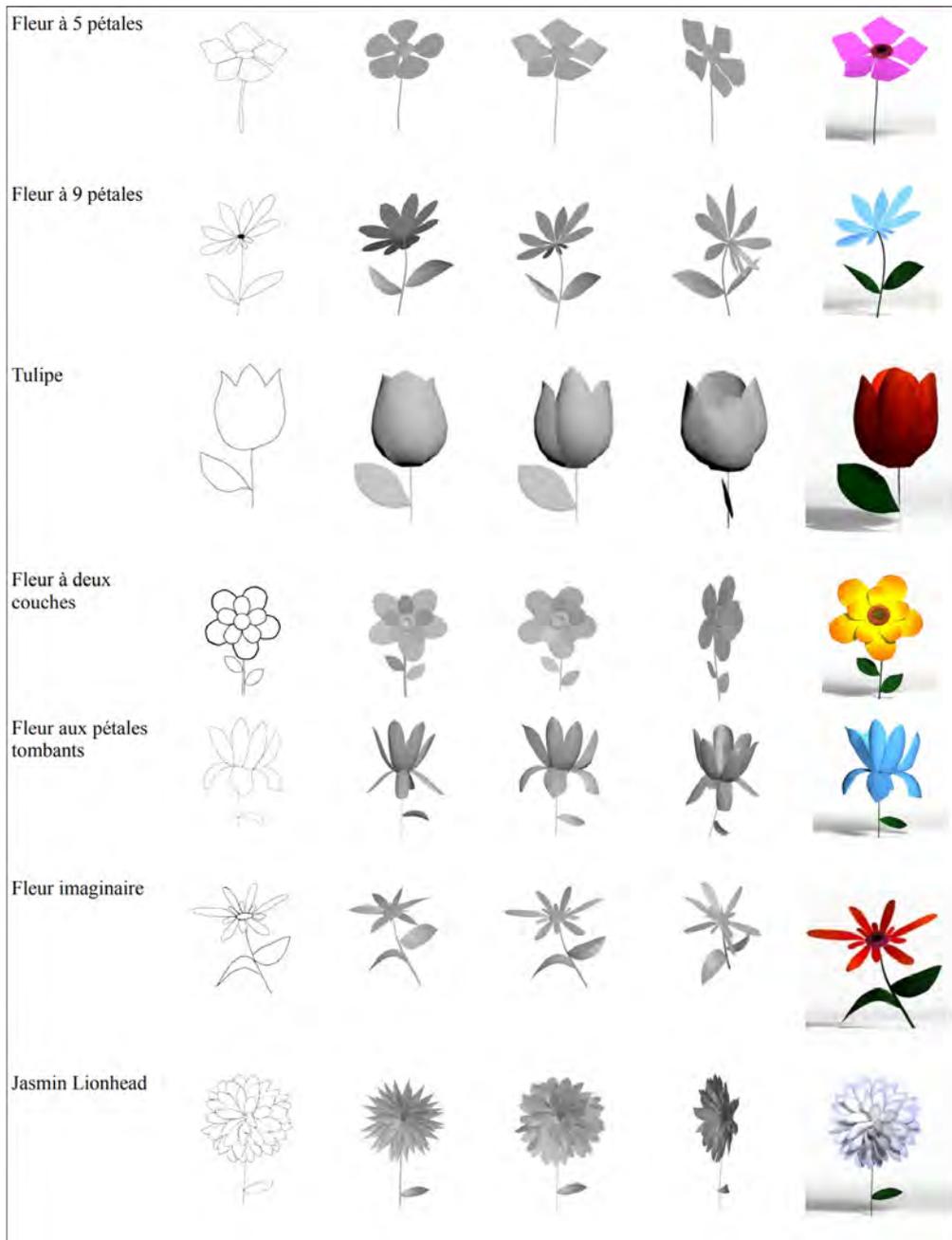


FIGURE 3.17 – Différents résultats obtenus avec notre modèleur. De gauche à droite : le croquis d'entrée réalisé par l'utilisateur ; le modèle 3D reconstruit en assemblant les différents éléments floraux récupérés dans la base de données uniquement ; le modèle final avec la déformation appliquée selon le même point de vue que sur le dessin ; le modèle final déformé vu d'une direction différente et le modèle avec une texture appliquée manuellement.

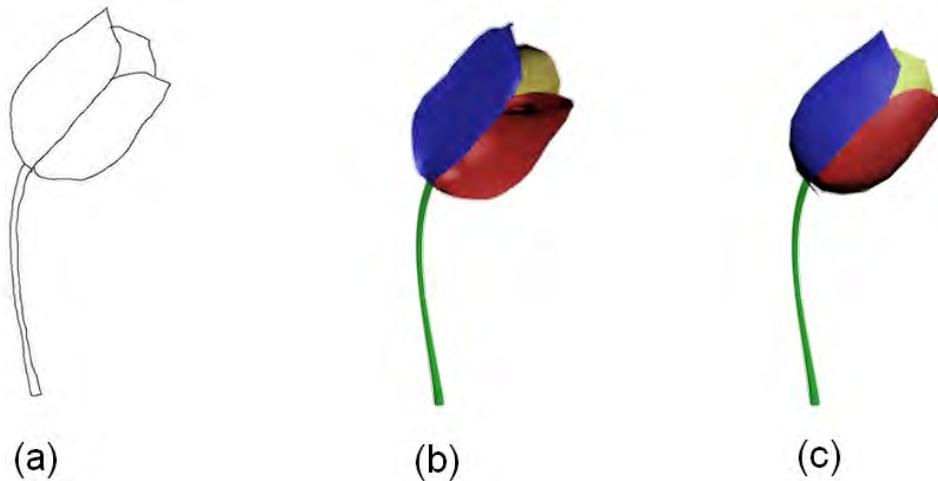


FIGURE 3.18 – Lorsque le dessin est dessiné selon une vue strictement de profil (a), les courbes des pétales segmentées ne retranscrivent pas la courbure nécessaire au modèle 3D. L'étape de récupération d'éléments floraux dans la base de données peut alors retourner des pétales correspondant parfaitement à la silhouette, mais complètement plats (b), alors que l'utilisateur désirerait des pétales avec une certaine courbure (c).

construction de fleurs. Notre méthode s'est avérée performante sur un vaste ensemble de croquis dessinés par plusieurs utilisateurs sans formation particulière, reconstituant avec succès des fleurs à une ou plusieurs couches, représenté selon différents angles de vue et avec différents degrés de floraison. De plus, elle fonctionne également pour la reconstruction de fleurs plus sophistiquées. Notre méthode tire parti du fait que les utilisateurs ont tendance à simplifier la complexité structurelle et géométrique des objets floraux dans leurs dessins 2D, ce qui nous permet de reconstruire un modèle réaliste correspondant aux intentions de l'utilisateur, malgré le fait qu'il ne soit pas correct botaniquement parlant (avec l'absence de certains éléments, tels que les pistils). Notre système profite également du fait que les fleurs possèdent une structure commune et que leur corolle peut être représentée par une forme géométrique relativement simple (un cône) dont les paramètres peuvent être calculés à partir des informations 2D présentes dans le dessin et les traits de guidage fournis par l'utilisateur. Par ailleurs, l'utilisation d'une base de données d'éléments floraux 3D permet d'assurer que le modèle 3D reconstruit soit réaliste. À notre connaissance, il s'agit du premier système de modélisation 3D à partir d'un croquis dédié à la reconstruction de fleurs qui est capable de produire des résultats d'une telle qualité et de manière robuste, qui ne requiert un dessin que d'une seule vue en donnée d'entrée, et avec une interaction avec l'utilisateur aussi réduite.

Comparée aux travaux déjà existants qui proposent une méthode de reconstruction pour les fleurs 3D, la nôtre offre de nombreux avantages. Tout d'abord, la modélisation

des fleurs avec notre système est extrêmement rapide et se résume à dessiner sa silhouette sur l'écran et de faire quelques traits de guidage supplémentaires, puis un résultat 3D est proposé en quelques secondes. Il est donc beaucoup plus efficace que d'autres méthodes interactives telles que [IOOI05], qui nécessitent 40 min pour la reconstruction d'un tournesol. Notre méthode propose également une liberté complète sur l'angle de vue choisi pour dessiner la fleur, ce qui n'est pas le cas de [YGCO<sup>+</sup>14] qui, bien qu'elle permette une reconstruction dans des temps similaires aux nôtres, est restreinte à des photos de fleur prises de face ou légèrement oblique uniquement. Un autre gros avantage est que notre méthode est réellement un outil basé sur l'esquisse. Le dessin, le guidage de l'utilisateur et donc la reconstruction, se font dans une seule vue, sans aucune manipulation interactive ni placement de pièces 3D sur la fleur (contrairement à [IOOI05]). Notre outil conserve donc la simplicité du dessin sur papier, augmentant sa facilité d'utilisation et le rendant accessible à presque tout le monde. Enfin, il permet de créer une variété de fleurs que tous les systèmes de modélisation ne peuvent pas forcément produire, y compris celles à plusieurs couches de pétales, et peut gérer un dessin en entrée plus ou moins abstrait ( tel que la tulipe, fig. 3.17).

Malgré tous ces avantages, notre système est loin d'être parfait, et des améliorations sont possibles. La première est évidemment de perfectionner la déformation. En effet actuellement cette étape n'est appliquée qu'aux pétales de la fleur, puisqu'il s'agit, selon nous, des éléments les plus importants de la fleur. Cependant, la tige ainsi que les feuilles, elles, ne sont pas déformées. Leur forme est donc limitée au contenu de la base de données, et peut ne pas correspondre parfaitement à l'esquisse de l'utilisateur. Il serait donc intéressant de voir comment l'on pourrait appliquer un processus de déformation à ces éléments également. La seconde amélioration possible serait d'implémenter la possibilité de gérer des dessins de couleurs, et d'utiliser cette information supplémentaire afin de générer et d'appliquer automatiquement des textures aux différents éléments floraux du modèle. À l'heure actuelle, la texturation de nos résultats 3D doit être faite manuellement. Une perspective pour ce travail serait également de voir s'il est possible de combiner notre système à une partie basée sur de l'apprentissage, notamment à l'aide de réseaux de neurones, qui permettrait de reconstruire les fleurs en surmontant les limites, et de notre méthode (géométrie restreinte aux modèles de la base de données) et de celle propre aux réseaux de neurones (le dessin en entrée est limité à l'angle de vue utilisé pour les données d'entraînement) utilisés par les méthodes de reconstruction actuelle telle que [HKYM17] et [LGK<sup>+</sup>17].

## Chapitre 4

# Descriptive : L'utilisation d'un dessin en style descriptif pour la modélisation géométrique de formes libres

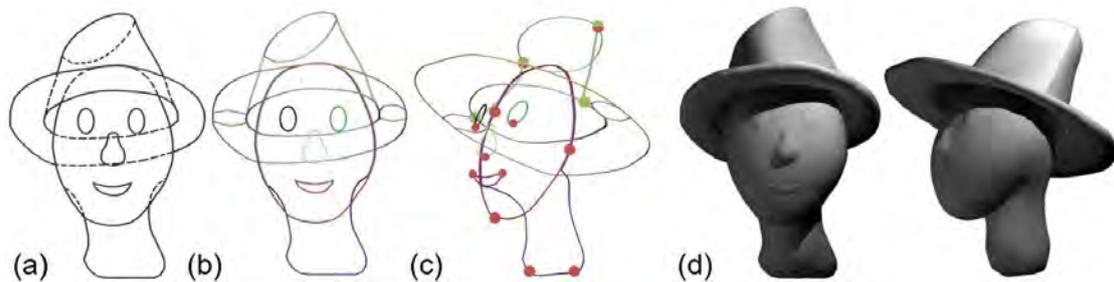


FIGURE 4.1 – Vue d'ensemble du processus de reconstruction avec notre modèleur. Le dessin fourni par l'utilisateur en style descriptif (a), l'utilisateur assemble des segments de courbes pour générer les courbes fermées correspondant aux limites des éléments fermés composant le modèle final (b). La position 3D des points de la courbe sont calculées (c) et la surface 3D est générée (d).

### 4.1 Introduction

Le chapitre précédent présente notre première contribution en proposant un modèleur dédié à la reconstruction d'objets floraux. Celui-ci permet aux utilisateurs de créer facilement et rapidement des fleurs 3D de bonne qualité à partir d'un seul croquis, sans nécessité des connaissances particulières en modélisation 3D ou dans l'utilisation de mo-

deleurs 3D classiques tels que 3DS Max ou Maya. Cependant, notre système fait face à une énorme limitation : les formes qu'il permet de créer possèdent une complexité et surtout une expressivité limitée, puisqu'il est restreint à la reconstruction d'un seul type d'objet, les fleurs.

Notre second modelleur a été développé en ayant pour objectif de ne pas souffrir de la même limitation, le but étant de proposer une solution au problème de modélisation géométrique d'une façon plus générale, c'est-à-dire en étant capable générer n'importe quelle forme 3D à partir d'un croquis d'un seul point de vue. La solution que nous proposons, nommé **Descriptive**, a été conçue avec cet aboutissement à l'esprit, et vise à élargir la gamme et la complexité des formes pouvant être créées à partir d'un croquis en vue unique. Notre système cherche également à limiter au maximum possible les interactions nécessaires avec l'utilisateur, de manière à conserver l'accessibilité pour tous que se doivent d'avoir les modelleurs basés sur des esquisses.

#### 4.1.1 Problématique

La principale difficulté à laquelle les modelleurs doivent faire face découle des ambiguïtés inhérentes à la reconstruction de formes 3D à partir d'un dessin 2D, c'est-à-dire au fait qu'il existe un nombre infini de formes 3D correspondant à un croquis donné. Ce problème devient encore plus difficile lorsque l'esquisse contient des parties cachées, donc non dessinées, qui doivent être estimées en 2D et en 3D. En effet, déterminer les éléments cachés est une tâche complexe à effectuer algorithmiquement, car cela implique d'avoir une compréhension sémantique du dessin. Par exemple, la partie cachée d'une forme donnée va dépendre du fait qu'elle représente un humain ou une plante. Une solution utilisée par plusieurs méthodes existantes est d'adopter des a priori sur le type de l'objet représenté de manière à avoir la compréhension nécessaire pour effectuer cette estimation. C'est d'ailleurs la solution que nous avons adoptée avec notre premier système dédié à la reconstruction des fleurs. Cependant, utiliser des a priori sur la nature de l'objet esquissé revient à limiter la reconstruction possible par le modelleur à ce type d'objet spécifiquement, et n'est donc pas une solution envisageable pour notre système. La problématique est donc la suivante : un croquis simple ne fournit pas les informations nécessaires pour effectuer une reconstruction 3D simplement, mais l'utilisation d'a priori, qui limiteraient le champ d'action du système, n'est pas possible.

De manière à résoudre ce problème il est donc nécessaire de trouver une autre source d'information, utilisable pour tous les modèles que l'utilisateur souhaite produire et qui ne limiterait pas la gamme de modèles restructurables. C'est en partant de ce constat que nous avons établi une stratégie en deux points qui est à la base de notre système : (1) puisqu'un croquis simple ne fournit pas les informations nécessaires, il suffit de fournir un croquis un peu plus informatif; et (2) si l'on a besoin de connaissances supplémentaires pour nous aider lors de la reconstruction, il suffit de les demander à l'utilisateur. Autrement dit, nous allons donner à celui-ci la possibilité de guider la

reconstruction 3D.

Notre modelleur, *Descriptive*, implémente donc cette stratégie. Il est ainsi nommé, car l'on demande à l'utilisateur de commencer par esquisser des courbes dites **descriptives**. Autrement dit, les éléments cachés sont également représentés sur le dessin, mais le sont avec des lignes en pointillés (fig. 4.1 (a)). Ceci nous permet non seulement de ne pas avoir besoin d'estimer les parties cachées, et donc de ne pas avoir besoin d'informations sémantiques sur le croquis, mais fournit également des informations partielles sur la position en profondeur des éléments composant l'objet les uns par rapport aux autres. Notre choix a été motivé par le fait que la représentation en pointillés est perçue naturellement par les artistes comme servant à décrire le non visible, principe notamment utilisé en géométrie descriptive, où les contours cachés sont dessinés par des courbes en pointillés. Une fois le croquis fourni, l'utilisateur peut guider la reconstruction automatique en spécifiant successivement des contraintes géométriques locales sur les points de courbe qu'il sélectionne. Ces interactions se font via une interface utilisateur simple et intuitive, qui reprend les conventions communément acceptées dans les outils de modélisation conventionnels. Ainsi, en combinant une esquisse d'une vue unique de l'objet, dans un style descriptif, avec la possibilité de spécifier des contraintes, nous obtenons un moyen efficace et flexible permettant à l'utilisateur de transmettre les informations nécessaires à la reconstruction 3D de n'importe quelle forme souhaitée.

Une autre difficulté à laquelle les modelleurs par croquis doivent faire face réside dans la complexité informatique inhérente aux problèmes d'optimisation qui sont utilisées pour exprimer le problème de reconstruction 3D. La modélisation 3D implique la nécessité de créer une surface de qualité, souvent représentée par un grand nombre de sommets. Or, tous ces sommets doivent être pris en compte dans le problème d'optimisation associé à la reconstruction 3D, et l'on se retrouve alors avec une tâche nécessitant un large de temps de calcul. Idéalement, la reconstruction doit être exécutée à des vitesses interactives afin que l'utilisateur puisse obtenir un retour visuel immédiat, ou au moins avec un temps d'attente négligeable. Pour ce faire, il est nécessaire que le problème d'optimisation soit exprimé de façon à être résolu efficacement. Dans *Descriptive* nous avons donc cherché à formuler ce problème comme une optimisation linéaire, afin de permettre la reconstruction avec des temps minimes.

La fig. 4.1 illustre le processus complet de modélisation avec notre système. L'utilisateur commence par dessiner le contour d'un objet, y compris les parties cachées, en utilisant le style de dessin descriptif (fig. 4.1 (a)). Comme nous l'avons précisé plus tôt, l'ambition à l'origine de *Descriptive* est d'avoir un modelleur qui n'aurait aucune limitation, ni dans le type d'objets à reconstruire ni dans le point de vue du croquis. L'utilisateur est donc libre de choisir la vue arbitraire de son choix pour le dessin d'entrée, bien que les concepteurs semblent préférer les points de vue qui exposent les multiples faces de l'objet et minimisent la variation entre la forme 3D et sa projection. Il est également

libre sur la nature de l'objet qu'il souhaite reconstruire, qui peut donc être organique ou artificiel, avoir des parties concaves ou convexes, etc.

Le dessin est ensuite partitionné en segments de courbes qui sont ensuite assemblés en courbes fermées ((fig. 4.1 (b)) correspondant chacune à la projection 2D d'un élément composant l'objet à reconstruire. Leurs équivalents 3D sont calculés pour permettre à l'utilisateur d'inspecter la forme 3D à reconstruire. Il peut, s'il le désire, ajouter des contraintes géométriques locales, que ce soit pour guider la reconstruction initiale ou pour modifier une reconstruction qui ne lui convient pas ((fig. 4.1 (c)). Dans tous les cas, une surface fermée est générée pour chaque courbe fermée de façon à ce que leur union donne une forme qui satisfasse à la fois les contraintes locales spécifiées par l'utilisateur et la contrainte projective, c'est-à-dire l'esquisse en entrée ((fig. 4.1 (d) ).

### 4.1.2 Travaux connexes

Descriptive partage de nombreuses idées avec des travaux préexistants. Le constat qu'un simple croquis n'apporte pas les informations suffisantes à la reconstruction n'est pas nouveau, et résoudre ce problème en améliorant la qualité informative de l'esquisse par l'utilisation d'un style de dessin plus descriptif est une idée qui a déjà été adoptée par des modeleurs existants. *True2Form* ([XCS<sup>+</sup>14]) et *BendFields* ([IBB15]) sont basés sur ce principe et utilisent des croquis de concepteur pour reconstruire des objets manufacturés. En effet, les concepteurs possèdent un style artistique particulier qui consiste à décrire complètement la géométrie de l'objet en traçant un ensemble de courbe supplémentaire, permettant ainsi de fournir des informations sur le parallélisme, l'orthogonalité et l'épaisseur de l'objet. De plus, ils représentent l'objet selon des angles de vue spécifiques, plus aptes à fournir le maximum d'information possible, faisant de ce type de dessin une donnée d'entrée parfaite pour les méthodes de reconstruction 3D. Ces systèmes souffrent malgré tout de quelques limitations. Les dessins réalisés dans le style utilisé par les concepteurs, comme le montre la fig. 4.2, sont très détaillés et répondent à des règles précises ; ces méthodes nécessitent donc que l'utilisateur soit capable de réaliser de tels croquis, ce qui ne les rend pas accessibles à tous. De plus elles sont limitées à la reconstruction d'objets manufacturés.

Au contraire, le style de dessin descriptif utilisé par Descriptive est extrêmement simple à prendre en main : aucun point de vue spécifique n'est à choisir, et la représentation des parties cachées en pointillés est naturelle ; rendant ainsi notre méthode accessible à tous, sans avoir besoin de connaissances artistiques préalables. Enfin ce style graphique n'est pas limité aux objets manufacturés, et Descriptive est également capable de reconstruire des formes organiques.

Notre modeleur se rapproche aussi de méthodes telles que *Teddy* ([IIMT07]), *SmooSketch* ([KH06]) ou *ShapeShop* ([SWSJ07]). En effet, tout comme pour ces systèmes, la reconstruction 3D dans Descriptive est effectuée par le biais d'une méthode d'inflation qui va gonfler dans les deux sens la région délimitée par les courbes du croquis. La valeur de ce gonflement, c'est-à-dire l'épaisseur qu'aura l'élément final, dépend de la largeur de

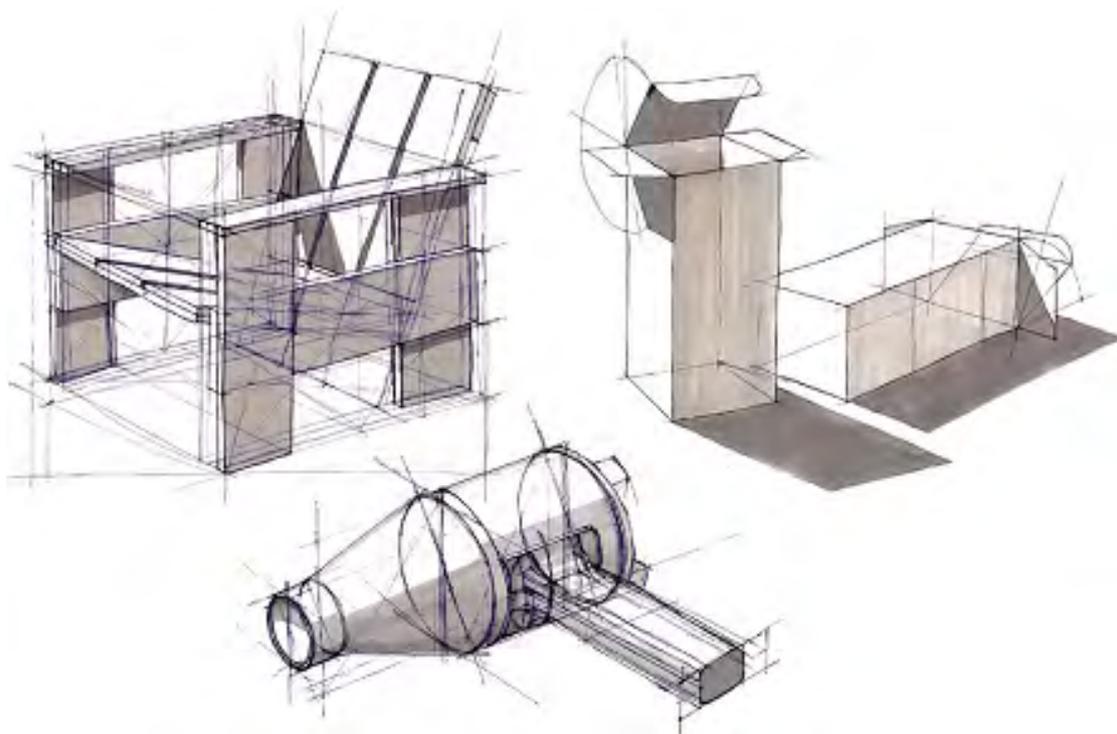


FIGURE 4.2 – Exemple de dessin de concepteur. Les éléments sont représentés par un ensemble de courbes dont le but est de transcrire parfaitement les propriétés géométriques de l’objet en question. Image de Alex Olieman, The Industrial Design Engineering Wiki.

la région : plus la région est large et plus l’élément sera épais, et inversement. Cependant ces modeleurs n’offrent qu’un contrôle extrêmement limité sur la forme reconstruite et notamment sur son épaisseur ; ils ne permettent alors pas, contrairement à *Descriptive*, de créer des objets très fins. Ils ne sont donc pas adaptés à la création d’objets manufacturés, comportant principalement des éléments plats, mais plutôt à la reconstruction de formes organiques. De plus, ils ne sont pas capables de créer facilement des objets complexes, tels qu’un modèle avec beaucoup d’entrelacement, ou un modèle composé d’un élément fin qui en englobe un autre (tel qu’un vêtement par dessus un corps par exemple). *Descriptive* se détache donc de ces méthodes grâce à l’ensemble d’outils simples qu’il propose à l’utilisateur et qui lui permettent d’avoir un contrôle complet de la forme reconstruite, lui permettant de produire de tels modèles complexes.

Enfin *Descriptive* peut également être rapproché des méthodes de reconstruction par annotations telles que *SKETCH* ([ZHH96]) ou *BendSketch* ([LPL<sup>+</sup>17]). Dans ces méthodes l’utilisateur apporte des informations nécessaires à la reconstruction par le biais d’annotations sur le dessin qui seront interprétées par le système comme instruction pour la reconstruction. L’ensemble des outils qu’offre *Descriptive* à l’utilisateur pour le

contrôle de la forme reconstruite peut, d'une certaine façon, être considéré comme des annotations puisque ces outils sont directement utilisables sur le dessin 2D. Cependant, Descriptive offre deux avantages par rapport aux méthodes par annotation classiques. Tout d'abord les interactions avec l'utilisateur sont minimales, et non obligatoires. De ce fait, le nombre d'*annotations* utilisées dans Descriptive est minimal en comparaison avec les autres méthodes, le rendant plus simple d'utilisation. Enfin, les outils utilisés dans Descriptive portent directement sur les sommets des courbes du croquis, et ne consistent pas en un ensemble de courbes supplémentaires qui doivent être esquissées par-dessus le croquis de l'utilisateur. Par conséquent, notre modéleur ne souffre pas de l'existence de la distance entre le croquis d'entrée, avec les annotations, et le résultat final que l'on reproche aux méthodes par annotation (section 2.1).

### 4.1.3 Contribution

La stratégie de modélisation mise au point, et utilisée par Descriptive, étend considérablement la puissance de modélisation des autres modéleurs basés sur des croquis, tout en conservant une interface intuitive et facile à utiliser. Comparé aux autres méthodes, Descriptive n'est limité à aucun type d'objet, faisant de lui un des modéleurs permettant la reconstruction de la plus large variété d'objets possibles. De plus, le fait que la reconstruction soit guidée par l'utilisateur nous permet d'avoir une méthode robuste. Contrairement aux méthodes automatisées, qui vont échouer à fournir une reconstruction si une de leurs étapes ne marche pas correctement, Descriptive pourra toujours fournir un résultat. Et si celui-ci ne correspond pas aux attentes de l'utilisateur, ce dernier pourra le corriger sans difficulté.

Point important, bien que Descriptive permette une liberté de reconstruction équivalente à celle proposée par les modéleurs classiques, tels que 3DS Max ou Maya, il reste beaucoup plus simple à prendre en main. En effet, il ne nécessite qu'une vingtaine de minutes d'entraînement avant d'être utilisable par n'importe qui, comme l'a montré notre étude utilisateur, et ne nécessite aucune connaissance préalable en modélisation 3D.

## 4.2 Interactions avec l'utilisateur

L'interface utilisateur de Descriptive a été développée en gardant à l'esprit l'un des points clés des méthodes de modélisation à l'aide de croquis : rendre la modélisation 3D accessible à tous. Il est donc nécessaire que l'interface utilisateur soit la plus facile à utiliser et soit conçue de manière à simplifier au maximum le processus de modélisation. Cette interface doit néanmoins permettre à l'utilisateur d'exprimer pleinement ces attentes et de créer une large variété de formes. De plus, les interactions entre l'artiste et le système doivent être minimales, de manière à ne pas rendre le processus de reconstruction trop lourd et trop complexe pour un utilisateur non-expert. Nous avons donc limité les moments où des interactions peuvent être nécessaires à 4 étapes. Tout d'abord, l'utilisateur commence par dessiner l'esquisse initiale (1) et regroupe les segments de courbe en un ensemble de courbes simples fermées (2) qui délimitent la surface des éléments

à reconstruire. Éventuellement, l'utilisateur peut ensuite modifier la forme calculée en spécifiant des contraintes sur certains points de courbes sélectionnés afin de guider sa reconstruction 3D (3). Il peut aussi modifier l'épaisseur le long de l'axe z des surfaces 3D générées pour chacun des éléments (4) et ainsi obtenir la forme finale désirée.

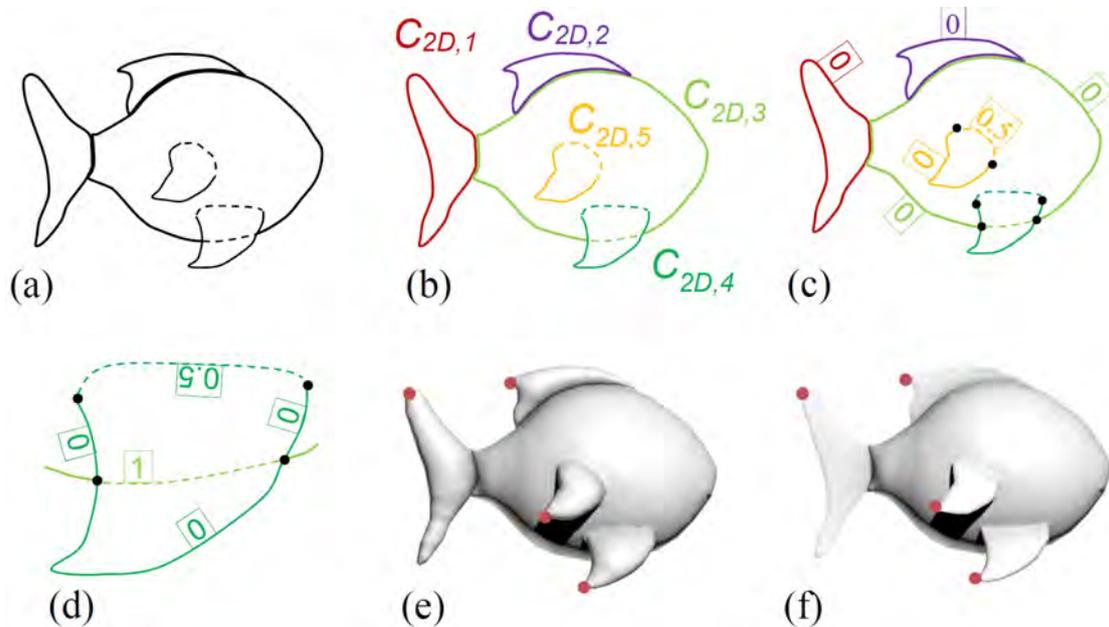


FIGURE 4.3 – L'utilisateur fournit un dessin initial. Les contours des parties cachées sont dessinés en pointillés (a) ; puis les segments de courbe du dessin sont décomposés en un ensemble de courbes fermées ( $C_{2D,1}$  à  $C_{2D,5}$ ) (b) ; l'indice de profondeur est calculé sur chaque section des courbes fermées (c) ; un zoom de la nageoire pelvienne pour afficher les indices de profondeur (c) ; des contraintes de profondeur facultatives (indiquées par des points rouges) peuvent être définies par l'utilisateur sur certains points sélectionnés sur la courbe ; des surfaces volumiques fermées sont ensuite créées en gonflant les patches de surface générés à partir des courbes fermées 3D(e), dont l'épaisseur du volume le long de l'axe z peut être ajustée par l'utilisateur (f).

**Dessin initial** Le dessin d'entrée est réalisé dans un style descriptif et fourni par l'utilisateur à l'aide d'une tablette. Les courbes y sont tracées de manière à représenter la silhouette ou les plis d'un objet selon un point de vue unique choisi arbitrairement par l'artiste (fig. 4.3 (a)). L'utilisateur dispose de deux outils de traçage différents, le premier pour représenter les éléments visibles et qui trace des traits pleins, le second qui permet de représenter les éléments cachés en lignes pointillées, en suivant le style d'esquisse de la géométrie descriptive. Le dessin est donc constitué par un ensemble de courbes polygonales sur le plan d'esquisse ( $z = 0$ ), visibles ou non, qui sont ensuite discrétisées automatiquement par le système.

**Segmenter le dessin en courbes fermées étiquetées avec leur indice de profondeur.** Une fois que le dessin est fourni par l'utilisateur, notre système va automatiquement diviser les courbes à chacune de leurs intersections, produisant ainsi de multiples segments de courbe. L'utilisateur va alors sélectionner et assembler des segments de courbes de manière à manuellement les grouper en un ensemble de courbes fermées simples (c'est-à-dire des courbes fermées sans auto-intersection). Lorsqu'un élément du dessin d'origine n'est pas représentable par une courbe fermée il est demandé à l'utilisateur d'ajouter des segments de courbe supplémentaires de manière à pouvoir décomposer cet élément en courbes fermées simples. Le résultat de cette segmentation est un ensemble de  $n$  courbes fermées simples  $C_{2D} = C_{2D, 1}, C_{2D, 2}, \dots, C_{2D, n}$  situées dans le plan d'esquisse ( $z = 0$ ) (fig. 4.3 (c)). Chaque segment de courbe se voit ensuite attribuer un indice de profondeur de manière semi-automatique. Cet indice, quand il sera calculé, correspondra au nombre de surfaces situées entre le point de vue et le segment de la courbe dans l'espace 3D. Il est utilisé pour définir l'ordre de profondeur correcte des courbes 3D, et donc celui que les éléments auront lors de l'étape de reconstruction. Initialement, les segments pleins de la courbe dans le dessin d'entrée se voient attribuer un indice de profondeur 0, puisque ces courbes représentent des éléments visibles, c'est-à-dire sans surface entre elle et l'observateur ; tandis que les segments en pointillés se voient attribuer la valeur de 1 par défaut. Les indices de profondeur seront par la suite mis à jour automatiquement pour positionner correctement chaque élément en 3D, en prenant en compte les intersections entre les éléments visibles et non visibles. Les courbes en pointillés correspondant à une jonction où un élément entre dans un autre, tel que la nageoire sur le corps d'un poisson (fig. 4.3 (c)), se verront attribuer un indice de profondeur composé d'une partie décimale en 0,5, retranscrivant ainsi le passage du visible (0) à l'invisible (1). Les indices de profondeurs peuvent également être ajustés manuellement par l'utilisateur lorsque l'ordre de profondeur calculé doit être redéfini. En effet, certains cas peuvent présenter une ambiguïté que le système ne peut résoudre par lui-même. Par exemple lorsque deux segments de courbe cachés se croisent, comme sur la (fig. 4.4 (a)), le segment de courbe correspondant à l'élément qui doit être situé derrière l'autre peut se voir affecté manuellement un ordre de profondeur de 2 ((fig. 4.4 (d-e)) en fonction de la forme voulue par l'utilisateur. Par défaut, une valeur de 1 est attribuée aux deux segments de courbe ((fig. 4.4 (b-c)), car les lignes en pointillés ne codent qu'un seul niveau d'occultation.

**Spécification de contraintes sur les points de la courbe.** Le fait de dériver des contraintes uniquement à partir de l'esquisse n'offre aucun contrôle à l'utilisateur sur la forme, et la reconstruction qui en découle pourrait ne pas correspondre aux attentes de l'artiste. C'est pourquoi, dans Descriptive, nous donnons la possibilité à l'utilisateur de guider la reconstruction 3D des courbes fermées à l'aide de spécifieurs, qui permettent de définir des contraintes locales sur un ou plusieurs points. De cette manière, l'utilisateur peut influencer sur la reconstruction comme il le désire. En utilisant un spécifieur de point d'angle, il peut préciser des points sur lesquels une discontinuité de la coordonnée  $z$ , et de la tangente à la courbe, peut avoir lieu, lui permettant ainsi d'indiquer manuel-

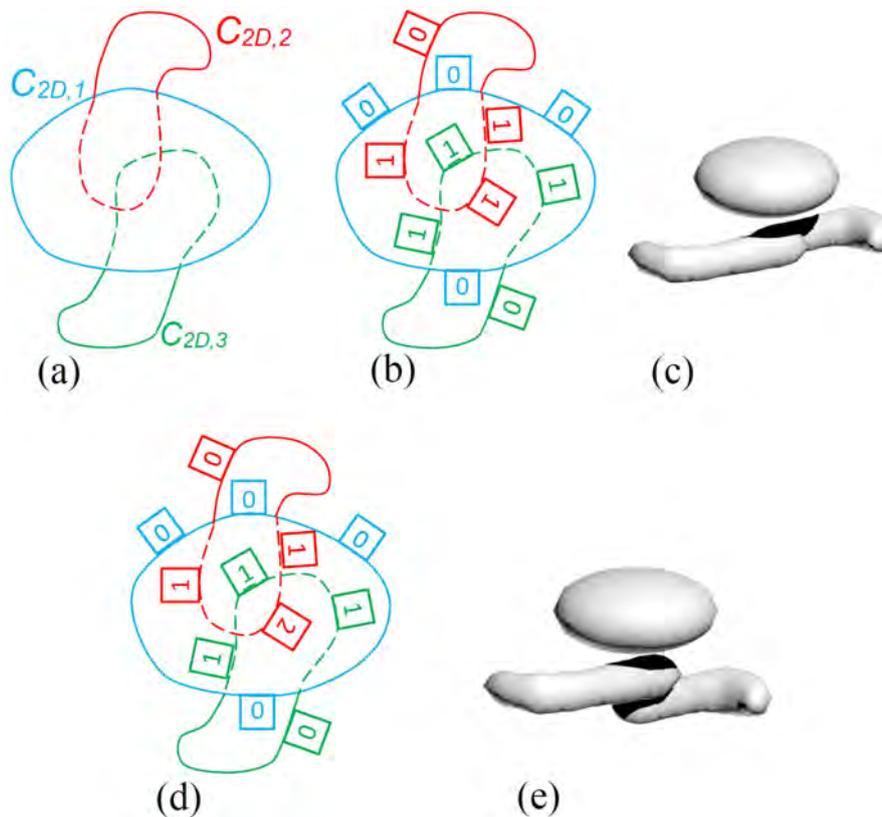


FIGURE 4.4 – Le croquis d’entrée (a) ; les courbes des éléments cachés se voient attribuer par défaut un indice de profondeur 1 (b) ; le résultat de la reconstruction montre que les parties de  $C_{2D, 2}$  et  $C_{2D, 3}$  se croisent puisqu’elles ont le même indice de profondeur (c) ; l’utilisateur peut attribuer manuellement un indice de profondeur de manière à placer  $C_{2D, 2}$  en dessous de  $C_{2D, 3}$  (d) ; le résultat de la reconstruction correspondante (e).

lement des coins (section 4.4.2). L’utilisateur peut également définir des contraintes de position pour spécifier directement la coordonnée  $z$  de certains points de courbe sélectionnés (points rouges de la fig. 4.3 (e)). Enfin, il peut également modifier la valeur par défaut du coefficient de planarité pour une courbe en particulier. Ce coefficient permet d’indiquer dans quelle mesure une courbe doit être plane en 3D au moment de la reconstruction. Par exemple, il peut ajuster ce coefficient pour faire varier l’orientation de surfaces entrelacées de manière à indiquer qu’une surface doit être moins déformée qu’une autre, afin d’éviter une interpénétration entre elles, qu’une autre (fig. 4.8). En donnant un coefficient plus élevé à un élément qu’à l’autre, on indique lequel doit rester le plus plat possible. Par ailleurs, l’utilisateur peut appliquer ces spécificateurs directement sur les courbes 2D du dessin initial, pour donner une indication dès la première reconstruction 3D, mais il peut également les utiliser après le positionnement des courbes en

3D, ayant ainsi la possibilité de pivoter la vue dans l'espace 3D. De cette manière, l'utilisateur peut modifier de manière interactive des courbes 3D ou des maillages de surface à partir de points de vue différents. Un point important à préciser est que le processus de modification du maillage 3D à l'aide de spécifieurs peut-être répété autant de fois que l'utilisateur le désire, le résultat 3D étant modifié à chaque spécification de nouvelles contraintes. Notez que ces interactions avec le système sont relativement rapides, ne nécessitant que quelques clics, et offre à l'utilisateur un contrôle total pour modifier le résultat 3D de manière à obtenir le résultat désiré.

**Réglage local de l'épaisseur des éléments de surface fermés.** Une fois que les courbes fermées 2D ont été positionnées dans l'espace 3D, le système génère à partir d'elles des éléments de surface fermés. Par défaut, un élément de surface a une section transversale circulaire (fig. 4.3 (e)). L'utilisateur a ensuite la possibilité de régler l'épaisseur de la surface le long de l'axe  $z$  globalement ou localement. De cette manière il lui est possible de créer des formes plates telles qu'une ailette (fig. 4.3 (f)) ou des formes concaves telles qu'un vêtement (fig. 4.21 (e)).

### 4.3 Génération de surface à partir des courbes 3D

Une fois que les courbes 2D issues du dessin de l'utilisateur auront été placées dans l'espace 3D, le système s'en servira afin de générer un maillage polygonal fermé. Il est important de prendre en compte la façon dont la surface est produite avant même de s'intéresser au placement des courbes en 3D, car c'est l'épaisseur des éléments 3D qui détermine le placement des courbes 3D les unes par rapport aux autres, et c'est la méthode de génération de maillage qui permet de définir l'épaisseur de chaque élément.

La génération du maillage final se fait via un processus en 3 étapes. Dans un premier temps, un patch de surface est calculé par interpolation à partir de chaque courbe. Cette interpolation consiste à créer de nouveaux sommets à l'intérieur de la région délimitée par la courbe, de manière à produire un patch à partir de celle-ci (fig. 4.5 (b)). Ce patch est ensuite dupliqué de manière à ce que l'un devienne la face avant de l'élément et l'autre la face arrière. Chacune de ces faces subit ensuite une étape d'inflation, qui consiste en un déplacement des sommets intérieurs du patch selon l'axe  $Z$ , permettant ainsi l'obtention d'éléments volumiques (fig. 4.5 (c)). Enfin, tous ces éléments de surfaces obtenus à partir de chaque courbe 3D sont fusionnés pour obtenir le modèle final, auquel est ensuite appliquée une opération de lissage.

**Génération du patch de surface** Définissons  $C_{3D,i}$  comme étant la courbe 3D composée de  $m$  points  $P_1, P_2, \dots, P_m$  et  $C_{2D,i}$  son équivalent 2D, c'est-à-dire une courbe simple fermée sur le plan de l'esquisse ( $z=0$ ). Dans un premier temps on effectue une triangulation de la région 2D délimitée par  $C_{2D,i}$  en utilisant la librairie **Triangle** de [She96] pour générer un maillage planaire  $M_{2D}$ . Précisons que les seuls sommets composant ce maillage 2D sont les sommets de la courbe  $C_{2D,i}$ . Cette triangulation dans

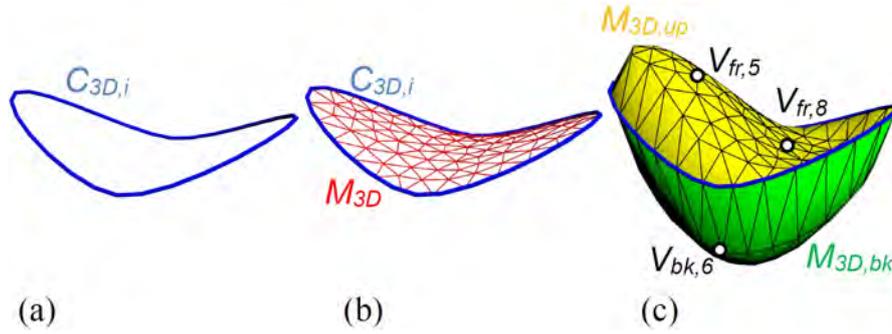


FIGURE 4.5 – Pour chaque courbe 3D  $C_{3D,i}$  (a), un patch de surface est généré par interpolation (b). En (c), ce patch de surface est dupliqué ( $M_{3D,fr}$  et  $M_{3D,bk}$ ), et chacun de ces patches est gonflé pour générer une surface volumétrique. L'épaisseur de ces surfaces est définie par des coefficients sur certains points du patch. Les coefficients de  $V_{fr,5}$ ,  $V_{fr,8}$  et  $V_{bk,6}$  ont été définis à 1,0, -0,03 et 1,0 respectivement.

le domaine 2D est ensuite mappée en 3D afin de créer un premier patch de surface 3D  $M_{3D}$ , une version dans l'espace, et donc levée, de  $M_{2D}$ . Cependant, actuellement, les seuls sommets de cette surface 3D sont les points limites de  $M_{3D}$ , c'est-à-dire ceux qui composent la courbe  $C_{3D,i} = P_1, P_2, \dots, P_m$ . Les sommets intérieurs  $V = V_1, V_2, \dots, V_o$  sont ensuite créés à l'intérieur de la région délimitée par  $C_{3D,i}$  en utilisant le calcul par valeur moyenne [HF06] dans le domaine 2D délimité par  $C_{2D,i}$ . La coordonnée  $z$   $Z_{V,j}$  d'un sommet intérieur  $V_j$  est ensuite définie comme :

$$Z_{V,j} = \sum_{k \in n} (W_k \cdot Z_{P,k}) \quad (4.1)$$

Où  $Z_{P,k}$  est la coordonnée  $z$  du point  $P_k$  de la courbe  $C_{3D,i}$ ,  $W_k$  est le poids correspondant, et  $n$  est le nombre de points de la courbe utilisée pour le calcul de la coordonnée moyenne. De cette opération résulte un patch de surface  $M_{3D}$  lisse couvrant l'intégralité de la région délimitée par la courbe  $C_{3D,i}$ .

**Duplication et inflation des patches** Le patch  $M_{3D}$  est ensuite dupliqué de manière à générer les deux faces qui composent l'élément volumique, donnant ainsi une face avant  $M_{3D,fr}$  et une face arrière  $M_{3D,bk}$ , qui sont cousus (soudés) le long de leur limite, c'est-à-dire le long de  $C_{3D,i}$ .

Chacun des sommets intérieurs de ces deux patches est ensuite déplacé le long de l'axe  $z$  afin de créer une surface volumétrique. Par défaut, le déplacement en  $z$  de chaque sommet est symétrique par rapport à la surface du patch et dépend de la distance entre le sommet et le centre du cercle circonscrit du triangle le contenant. Plus cette distance est grande, et plus le déplacement sera faible (jusqu'à être nul à la limite de la surface). De cette manière le résultat est un élément volumique symétrique dont la section transversale est circulaire.

Suite à la duplication du patch nous pouvons alors définir  $V_{fr,j}$  et  $V_{bk,j}$  comme étant deux sommets intérieurs appartenant respectivement à la face avant et la face arrière du patch  $M_{3D}$  et qui partagent les mêmes coordonnées  $x$  et  $y$ , et ont une position en  $z$  initiale commune. Cette position est ensuite modifiée par un déplacement le long de l'axe  $z$  calculé de la façon suivante :

$$Z_{V,fr,j} = Z_{V,j} + \beta_{V,fr,j} \cdot d_{V,j} \quad (4.2)$$

$$Z_{V,bk,j} = Z_{V,j} - \beta_{V,bk,j} \cdot d_{V,j} \quad (4.3)$$

où

- $Z_{V,j}$  est la coordonnée  $z$  initiale des points, et correspond à la coordonnée du sommet  $V_j$  sur le patch de surface.
- $d_{V,j}$  désigne la distance au patch de surface le long de l'axe  $z$  au sommet  $V_j$ , c'est-à-dire l'épaisseur que l'élément doit avoir. Pour calculer cette valeur, une Triangulation de Delaunay sous contrainte (CDT) de  $C_{2D,i}$  est d'abord effectuée dans le domaine  $x,y$ . Ensuite, pour chaque  $V_j$  nous trouvons le triangle  $T_j$  contenant dans le domaine  $x,y$  (fig. 4.6 (a)). De manière à calculer  $d_{V,j}$  nous définissons d'abord  $O_j$  et  $r_j$  comme étant respectivement le centre et le rayon du cercle circonscrit de  $T_j$  (fig. 4.6 (b)).  $d_{V,j}$  peut ensuite être obtenu par la formule  $d_{V,j} = \sqrt{r_j^2 - \|O_j - V_j\|^2}$
- $\beta_{V,fr,j}$  et  $\beta_{V,bk,j}$  sont les coefficients qui servent à contrôler localement l'épaisseur de l'élément. Par défaut, ils ont une valeur de 1, mais l'utilisateur peut les modifier pour les sommets de son choix. Lorsque l'utilisateur modifie la valeur d'un sommet, les coefficients de tous les autres sommets de la même face sont interpolés à l'aide des fonctions de base radiales (RBF) [Buh03] en utilisant les coefficients spécifiés par l'utilisateur comme interpolant. Notons que pour ces fonctions nous avons utilisé la distance euclidienne dans le domaine  $x,y$  ainsi que le noyau gaussien, la variance de chaque noyau étant fixée à deux fois la distance au plus proche voisin.

Par ailleurs, lorsque l'utilisateur ne spécifie un coefficient que pour un seul sommet, celui-ci est alors utilisé pour tous les sommets du patch. De cette manière l'utilisateur peut modifier l'épaisseur globale de l'élément en ne spécifiant qu'une valeur. Il est important de préciser également que les coefficients d'épaisseur sont déterminés pour chaque côté indépendamment, ce qui permet la création d'un élément volumique avec des épaisseurs, et donc des formes, différentes de chaque côté. Un coefficient d'épaisseur négatif peut également être précisé par l'utilisateur de manière à générer une surface concave ( $V_{fr,8}$  dans la fig. 4.5). Il peut également créer un élément complètement plat en définissant le coefficient avant et arrière à 0. Des exemples de telles formes sont représentés sur la fig. 4.21 (c) et (i) respectivement.

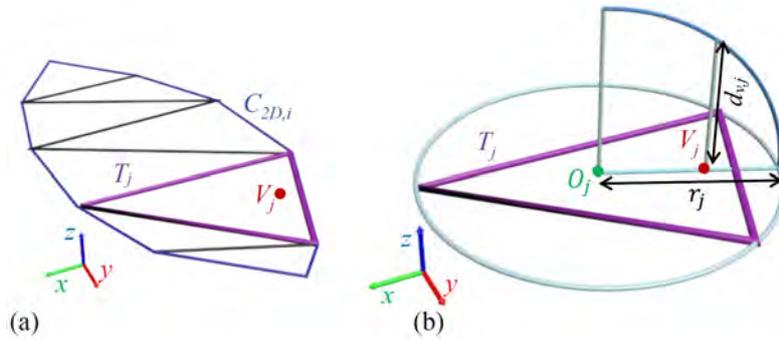


FIGURE 4.6 – Dans un premier temps, le triangle  $T_j$  contenant  $V_j$  est d’abord identifié (a). Ensuite, l’épaisseur  $d_{V,j}$  en  $V_j$  est calculée à l’aide du centre  $O_j$  et du rayon  $r_j$  du cercle circonscrit de  $T_j$ . Ce cercle circonscrit est représenté en bleu clair.

## 4.4 Reconstruction 3D des courbes fermées 2D

En sachant la façon dont la surface est générée, et donc l’épaisseur que chaque élément doit avoir, il est alors possible de déterminer la position 3D que les courbes simples fermées doivent avoir, une fois leur indice de profondeur déterminé. En partant d’un ensemble de  $m$  courbes simples fermées  $C_{2D} = C_{2D, 1}, C_{2D, 2}, \dots, C_{2D, m}$  sur le plan d’esquisse ( $z = 0$ ) et dont l’ordre de profondeur a été déterminé, nous procédons au calcul des courbes 3D correspondantes  $C_{3D} = C_{3D, 1}, C_{3D, 2}, \dots, C_{3D, m}$ . Pour ce faire, nous cherchons à soulever les courbes  $C_{2D, j}$  du plan d’esquisse. Puisque le dessin est une projection orthogonale du modèle 3D, les coordonnées  $x, y$  des points des courbes du dessin sont les mêmes que ceux des courbes 3D correspondantes et seules leurs coordonnées  $z$  sont à calculer. Ceci est réalisé en résolvant un problème d’optimisation qui va chercher l’emplacement optimal des courbes 3D  $C_{3D, j}$ , tout en respectant un ensemble de contraintes d’égalité et d’inégalité qui servent à assurer que les éléments 3D reconstruits respectent l’ordre défini par les indices de profondeur des courbes 2D. Ce chapitre va expliquer plus en détail comment est exprimé ce problème d’optimisation. La fonction objective est décrite section 4.4.1, ainsi que les deux types de contraintes (section 4.4.2 et section 4.4.3) qui permettent le calcul des courbes 3D fermées (section 4.4.4).

### 4.4.1 Fonction objective de planarité

Nous calculons les courbes 3D fermées  $C_{3D, j}$  qui décrivent le mieux les contours de la forme voulue via une optimisation. Cela consiste à calculer la coordonnée  $z$  de chaque point des courbes fermées 2D, discrétisées à l’aide d’un algorithme d’échantillonnage de courbe standard. Le critère principal utilisé pour cette optimisation est la planarité locale des éléments. Nous avons décidé d’utiliser ce critère, car il peut être considéré comme une mesure de la qualité de lissage d’une courbe. Notre but étant de reconstruire un maillage lisse, nous avons besoin d’utiliser un critère qui assure que la solution

retournée par l'optimisation le soit également. Notre modeler va donc essayer de trouver les coordonnées  $z$  de manière à ce que les points voisins soient le maximum possible dans un même plan 3D. Des critères similaires ont été adoptés par d'autres systèmes, tels que l'invariance affine locale dans [XCS<sup>+</sup>14] ou la minimisation de courbure [CS07].

Chercher la planarité locale consiste à chercher la planarité à la plus petite échelle possible, c'est-à-dire faire en sorte que 4 points consécutifs de la courbe se situent dans un même plan. Définissons donc  $(p_i, p_{i+1}, p_{i+2}, p_{i+3})$  comme étant 4 points consécutifs de  $C_{3D,j}$  et  $(x_i, y_i, z_i)$ ,  $(x_{i+1}, y_{i+1}, z_{i+1})$ ,  $(x_{i+2}, y_{i+2}, z_{i+2})$  et  $(x_{i+3}, y_{i+3}, z_{i+3})$  leurs coordonnées respectives. Le problème des moindres carrés permettant d'obtenir le plan le plus adéquat passant par ces 4 points ( $z = a \cdot x + b \cdot y + c$ ) est :

$$\min_{Q_{i:i+3}} \|P_{i:i+3} \cdot Q_{i:i+3} - Z_{i:i+3}\|^2 \quad (4.4)$$

où

$$P_{i:i+3} = \begin{pmatrix} x_i & y_i & 1 \\ x_{i+1} & y_{i+1} & 1 \\ x_{i+2} & y_{i+2} & 1 \\ x_{i+3} & y_{i+3} & 1 \end{pmatrix}$$

correspond à la matrice des coordonnées  $x$  et  $y$  des 4 points consécutifs, connues puisqu'elles sont identiques aux coordonnées 2D,

$$Z_{i:i+3} = (Z_i \ Z_{i+1} \ Z_{i+2} \ Z_{i+3})^T$$

correspond aux coordonnées  $z$  que nous cherchons, et

$$Q_{i:i+3} = (a_{i:i+3} \ b_{i:i+3} \ c_{i:i+3})^T$$

étant les coefficients du plan passant par ces points.

Les variables non connues dans le problème de minimisation ci-dessus (eq. (4.4)) sont les coefficients du plan  $Q_{i:i+3}$ , car il s'agit d'un problème cherchant un plan passant par des points donc les coordonnées  $(x,y,z)$  sont connues. Cependant, dans notre cas, ce que nous cherchons à trouver est les coordonnées  $z$  des points. Nous devons donc réécrire cette équation de manière à ce que l'inconnu soit  $Z_{i:i+3}$ . Pour ce faire nous partons de la solution de eq. (4.4) qui est :

$$Q_{i:i+3} = (P_{i:i+3}^T \cdot P_{i:i+3})^{-1} P_{i:i+3}^T \cdot Z_{i:i+3}$$

De plus, on peut définir le vecteur  $Z'_{i:i+3}$  contenant les coordonnées  $z$  des 4 points qui se situent sur le plan donné par la solution à la minimisation précédente comme étant :

$$Z'_{i:i+3} = P_{i:i+3} \cdot Q_{i:i+3}$$

Ce qui revient à

$$Z'_{i:i+3} = P_{i:i+3} \cdot ((P_{i:i+3}^T \cdot P_{i:i+3})^{-1} P_{i:i+3}^T \cdot Z_{i:i+3})$$

On peut alors reformuler le problème de minimisation avec, comme inconnue, le vecteur  $Z_i$  contenant les coordonnées  $z$  des 4 points :

$$\min_{Z_{i:i+3}} \|Z_{i:i+3} - Z'_{i:i+3}\|^2$$

que l'on peut réécrire en factorisant  $Z_{i:i+3}$  :

$$\min_{Z_{i:i+3}} \|R_{i:i+3} \cdot Z_{i:i+3}\|^2 \quad (4.5)$$

$$\text{avec } R_{i:i+3} = (1 - P_{i:i+3}(P_{i:i+3}^T P_{i:i+3})^{-1} P_{i:i+3}^T).$$

Le point important est que cette nouvelle matrice  $R_{i:i+3}$  n'est pas calculée à partir des coefficients du plan ; il s'agit d'une matrice carrée qui ne dépend que des coordonnées  $x,y$  des 4 points  $(P_i, P_{i+1}, P_{i+2}, P_{i+3})$ , c'est-à-dire qu'elle ne dépend que de valeurs que nous connaissons directement par le dessin. Notons que cette matrice peut être mal conditionnée, notamment lorsque les 4 points sont parfaitement alignés. Dans ce cas, nous modifions légèrement les coordonnées  $x,y$  des points en ajoutant un bruit aléatoire minimale de manière à ce que l'alignement soit brisé et que  $R_{i:i+3}$  se retrouve correctement conditionnée.

La fonction objective de notre problème d'optimisation peut alors être définie pour les coordonnées  $z$  de tous les points de  $C_{3D,j}$  comme étant l'énergie de planarité :

$$\min_Z \|R_{C_{3D,j}} \cdot Z\|^2 \quad (4.6)$$

où  $Z$  correspond au vecteur contenant l'ensemble des coordonnées  $z$  des  $n$  points de toutes les courbes  $C_{3D}$  :

$$Z = (Z_1 \ Z_2 \ \dots \ Z_n)^T$$

Quant à elle, la matrice  $R_{C_{3D,j}}$  est remplie avec l'ensemble des éléments des matrices  $R_{i:i+3}$  calculées pour tout les 4 points consécutifs de  $C_{3D,j}$  en partant du point  $P_i$ .

Une attention particulière doit être portée lorsque deux courbes se croisent ou se chevauchent, partageant un ou plusieurs points consécutifs en 2D. Lorsqu'un élément est partiellement caché par un autre, leurs courbes en 2D passent généralement par un seul point 2D commun aux deux courbes (jonction en T), mais qui correspond en réalité à deux points 3D distincts. Dans ce cas, les points d'intersection sont dupliqués afin qu'ils puissent avoir des coordonnées  $z$  différentes sur chaque courbe 3D ( $P_6$  et  $P_{12}$  sur la fig. 4.7). Par contre, lorsque deux courbes partagent un ensemble de points consécutifs, par exemple lorsqu'elles se joignent à une jonction en Y, nous considérons que les arêtes des éléments de surface 3D correspondants sont attachées le long des points partagés, et donc le long de ces arêtes ( $P_1$  et  $P_2$  sur la fig. 4.7). Cela implique que ces points partagent les mêmes coordonnées  $z$  3D entre ces deux courbes, et ils ne sont alors pas dupliqués. La fig. 4.7 illustre un exemple de calcul de matrices  $R_{C_{3D,1}}$ , et  $R_{C_{3D,2}}$  pour deux courbes  $C_{3D,1}$  et  $C_{3D,2}$  se chevauchant et formant une jonction en T et deux jonctions en Y.

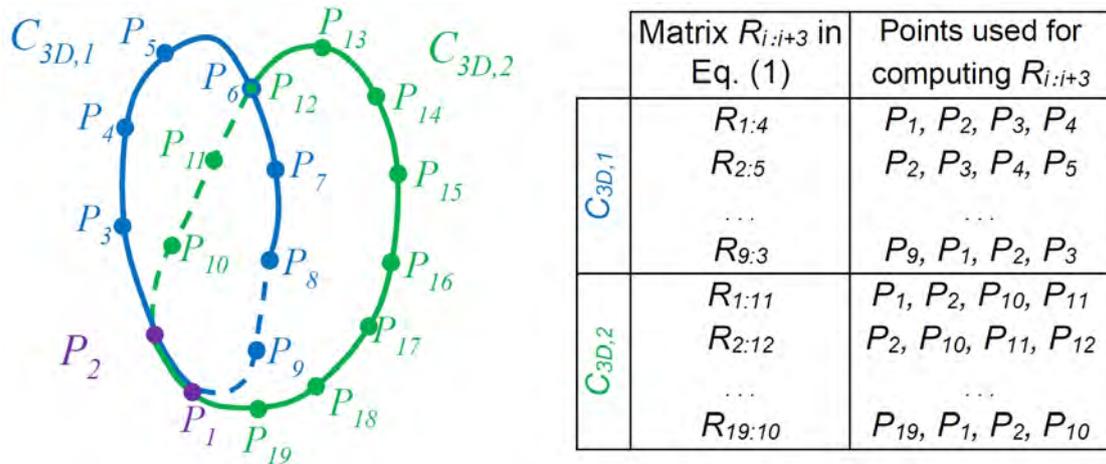


FIGURE 4.7 – Les matrices  $R_{i:i+3}$  sont calculées dans l'équation (2) pour tous les points de  $C_{3D,1}$  et  $C_{3D,2}$ .  $C_{3D,1}$  et  $C_{3D,2}$  ont les sommets  $P_1$  et  $P_7$  en commun.

La fonction objective est alors définie comme étant la somme de l'énergie de planarité (eq. (4.6)) de toutes les courbes présentes sur le dessin. Cependant, nous avons expliqué section 4.2 que nous voulions permettre à l'utilisateur d'avoir une certaine liberté d'expression sur la reconstruction 3D, et que cela passait notamment par la possibilité de renforcer ou d'affaiblir la planarité de certains éléments. Nous avons donc modifié la fonction objective de manière à pouvoir influencer sur l'impact de chaque courbe. Elle est alors définie pour les  $m$  courbes  $C_{3D}$  comme étant une somme pondérée :

$$E_{planarity} = \min_Z \left\| \sum_{j=1}^m \alpha_j \cdot R_{C_{3D,j}} \cdot Z \right\|^2 \quad (4.7)$$

où le coefficient  $\alpha_j$  est le poids ajustable manuellement par l'utilisateur, représentant l'importance de la planarité de  $C_{3D,j}$ . Plus la valeur donnée par l'utilisateur est grande, et plus la courbe restera dans le même plan. À l'inverse, plus la valeur donnée est petite, et plus la forme aura une liberté de réorientation locale. De cette manière l'utilisateur peut influencer la reconstruction en donnant plus d'importance à la planarité de la surface qu'il désire déformer le moins possible (fig. 4.8). Par défaut chaque courbe à la même importance avec une valeur de 1.

**Assemblage des éléments fermés.** Descriptive travail avec des courbes 2D fermées. Ce sont donc des éléments de surfaces simples qui sont reconstruits en 3D. Malgré tout il est cependant possible de créer des formes plus complexes, telles que des formes avec des trous, en décomposant l'élément complexe en multiples éléments simples qui partagent des zones en commun, c'est-à-dire qui se chevauchent partiellement. Durant l'étape de reconstruction, ces éléments seront assemblés permettant ainsi d'obtenir la forme désirée. La fig. 4.9 montre comment est modélisé un tore en utilisant deux courbes fermées

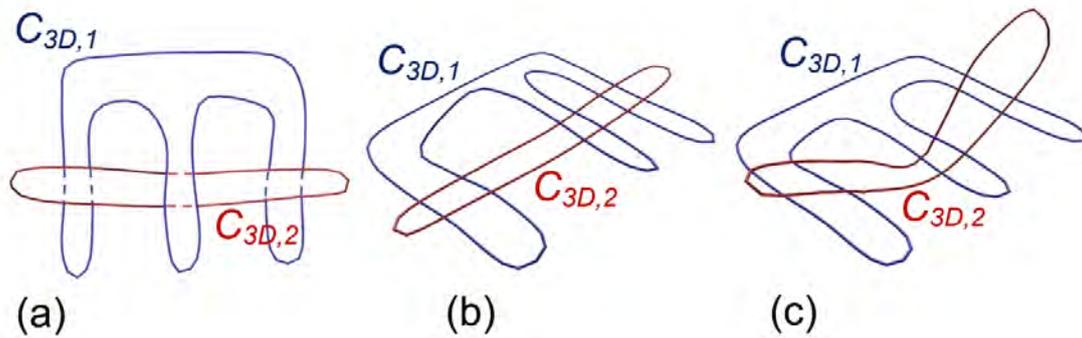


FIGURE 4.8 – Utilisation de différentes valeurs pour le poids de la planarité de chaque courbe. Les courbes dans le plan d’esquisse (a), la reconstruction avec  $\alpha_1 = 1$  et  $\alpha_2 = 10$  (b), la reconstruction des mêmes courbes, mais avec  $\alpha_1 = 10$  et  $\alpha_2 = 1$  (c).

partageant un ensemble de sommets consécutifs,  $P_i, P_{i+1}, P_{i+2}$  et  $P_{i+3}$ , ainsi que  $P_j, P_{j+1}, P_{j+2}$  et  $P_{j+3}$ .

La fonction objective de planarité étant appliquée aux deux courbes, elle va chercher à faire en sorte que ces points restent dans le même plan durant la reconstruction 3D. Comme les points ne sont pas alignés, il n’existe qu’un seul plan solution qui permet de s’assurer de la planarité pour ces 4 points, et il s’agit du même plan pour les deux courbes. Puisque les deux courbes se chevauchent, et que le plan dans lequel les points sont placés est également commun à ces courbes, les deux éléments 3D résultants seront collés ensemble, donnant l’illusion de ne former qu’un seul élément : le tore.

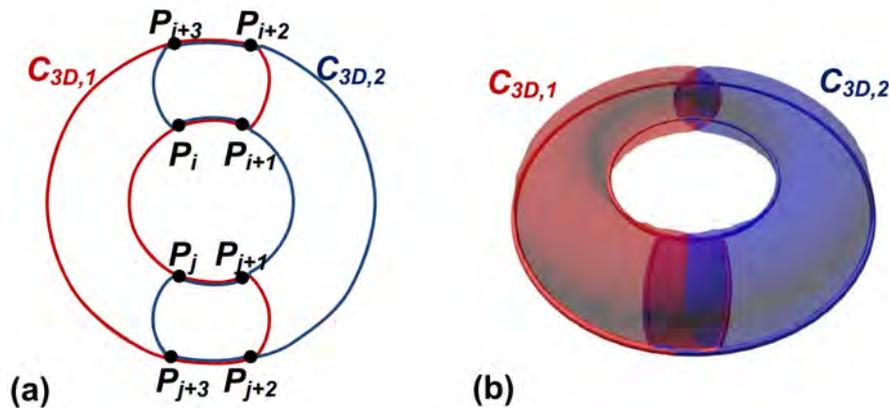


FIGURE 4.9 – La modélisation d’un tore est effectuée en le décomposant en deux courbes fermées se chevauchant  $C_{3D,1}$  et  $C_{3D,2}$ .

Il est également possible de créer des éléments de surface qui partagent une même bordure. De cette manière, notre système est capable, en assemblant des éléments possédant un ensemble de sommets en commun, de reconstruire des formes avec des crêtes, tel que le montre la figure 4.10, tout en respectant notre fonction objective de planarité. L'objet final est simplement décomposé en un ensemble d'éléments fermés correspondant chacun à un côté de la crête, c'est-à-dire que chacun se situe dans un plan différent de l'autre. De cette façon, chaque élément respecte la fonction objective, tout en permettant la création d'objets pourtant non planeaire.

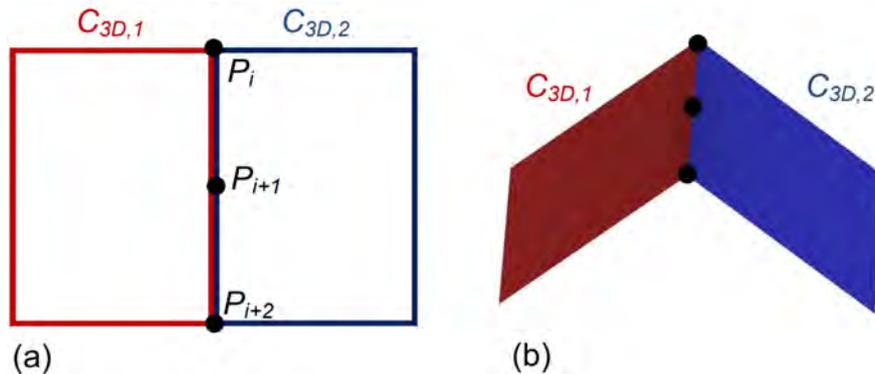


FIGURE 4.10 – La modélisation d'un toit (b) se fait en utilisant deux courbes (a). Ces deux courbes partagent un ensemble de points en commun  $P_i, P_{i+1}$  et  $P_{i+2}$ , qui sont alignés, formant ainsi le faîtage du toit 3D.

#### 4.4.2 Spécifieurs locaux

Notre objectif est de fournir un système permettant de reconstruire de manière automatique un objet complet à partir d'une esquisse unique, tout en laissant à l'utilisateur un certain contrôle pour obtenir la forme désirée. Nous mettons donc à leur disposition des spécifieurs locaux, qui leur permettent de guider ou modifier la reconstruction 3D initiale des courbes ou des surfaces. Ce chapitre va présenter les deux types de spécifieurs dont dispose l'utilisateur : les points d'angle et les contraintes de position.

##### Spécifieur de points d'angle

L'utilisateur peut sélectionner et appliquer sur certains points de son choix un spécifieur de point d'angle. Celui-ci aura pour effet de préciser que la règle de planarité ne doit pas s'appliquer à ce point. Notre fonction objective s'assure d'une planarité locale sur 4 points consécutifs. Ainsi, un même point apparaîtra dans 4 matrices de calcul de planarité, c'est-à-dire que pour le point  $P_i$  de la courbe, la règle de planarité va impliquer le calcul des 4 matrices  $R_{i-3:i}, R_{i-2:i+1}, R_{i-1:i+2}$  et  $R_{i:i+3}$  (eq. (4.5)) correspondant aux

ensembles de 4 points consécutifs commençant respectivement par  $P_{i-3}, P_{i-2}, P_{i-1}$  et  $P_i$  et contenant le point  $P_i$ .

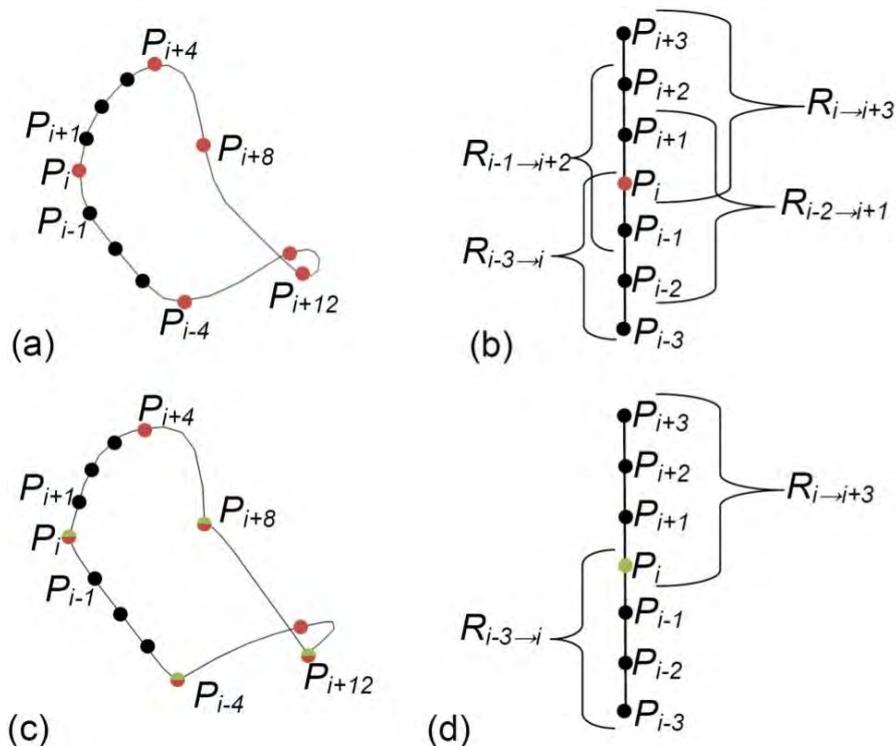


FIGURE 4.11 – La courbe  $C_{3D,i}$  positionnée en 3D avec des contraintes de position placées par l'utilisateur, en rouge, comme seul spécifieur utilisé(a), et les matrices calculées correspondantes  $R_{i-3:i}, R_{i-2:i+1}, R_{i-1:i+2}$  et  $R_{i:i+3}$  passant par le point  $P_i$  qui est une contrainte de position. La même courbe  $C_{3D,i}$  positionnée en 3D en plaçant les points d'angles aux points  $P_i, P_{i+8}, P_{i+12}$  et  $P_{i-4}$ , en plus des contraintes de position initialement placées (symbolisés par l'ajout de la couleur verte) (c), impliquant que les seules matrices calculées pour le point  $P_i$  sont les matrices  $R_{i-3:i}$  et  $R_{i:i+3}$ .

Lorsqu'un spécifieur de point d'angle est appliqué au point  $P_i$ , alors ce dernier n'est plus soumis à la règle de planarité, ce qui signifie que les seules matrices qui seront calculées sont celles où le point  $P_i$  est le premier ou le dernier des 4 points consécutifs. Seules les matrices  $R_{i-3:i}$  et  $R_{i:i+3}$  sont alors prises en compte pour calculer  $R_{C_{3D},j}$  dans la fonction objective (eq. (4.5)). Un point où est appliqué un spécifieur de point d'angle est alors un point où la coordonnée  $z$  peut changer brusquement, car la fonction objective ne va plus s'assurer que les voisins directs de ce point soient dans un même plan. Ce genre de points est utile, par exemple, pour créer des courbes 3D fermées constituées d'un ensemble de courbes planaires interconnectées. La fig. 4.11 montre un exemple d'une

courbe 3D composée de 4 courbes planes en utilisant 4 spécifieurs de point d'angle. Par ailleurs, si deux points d'angle sont spécifiés par l'utilisateur de manière qu'il n'existe qu'un seul point  $p$  présent entre eux, alors celui-ci ne sera pas impliqué du tout dans la règle de planarité. Puisqu'il est nécessaire d'avoir 4 points consécutifs pour définir une matrice de planarité, mais que des points d'angle ne peuvent être que le premier ou le dernier de ces 4 points consécutifs, alors il est impossible de calculer une matrice de planarité qui prendrait en compte ce point  $p$ .

### Spécifieur de contraintes de position

Le deuxième type de spécifieur dont dispose l'utilisateur est celui des contraintes de position. Comme le nom l'indique, il permet de définir manuellement la position que doit avoir le point sélectionné en indiquant la coordonnée  $z$  à laquelle il doit être placé. Ce type de spécifieur est utilisé pour placer les courbes 3D à une position  $z$  précise, ou pour forcer une courbe à prendre une forme non planeire tel que dans la fig. 4.12.

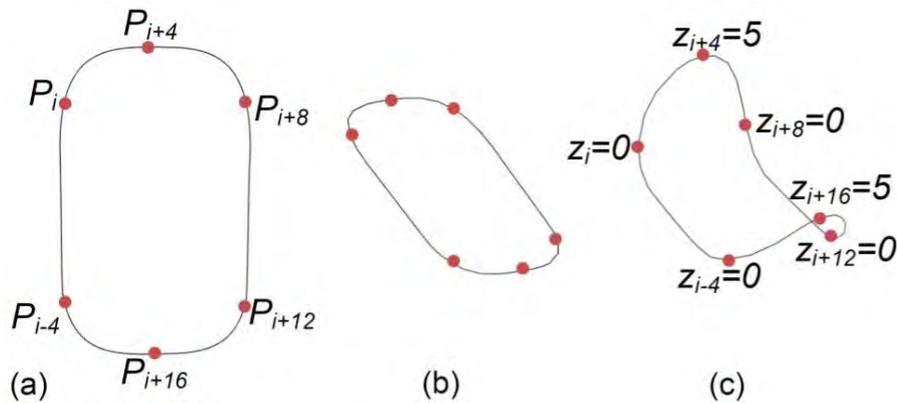


FIGURE 4.12 – La courbe  $C_{2D,i}$  (a), la courbe  $C_{3D,i}$  correspondante calculée en utilisant uniquement la règle de planarité (b), et la même courbe  $C_{3D,i}$  calculée en spécifiant des contraintes de position aux points  $P_i, P_{i+4}, P_{i+8}, P_{i+12}, P_{i+16}$  et  $P_{i-4}$  (c).

Ces contraintes de position consistent à fixer les points à une coordonnée  $z$ , elles sont donc écrites sous la forme d'un ensemble d'égalités :

$$\begin{cases} Z_i = Z_{C,i} \\ Z_j = Z_{C,j} \\ \vdots \end{cases}$$

où  $(Z_i, Z_j, \dots)$  correspondent à l'ensemble des coordonnées  $z$  des points de la courbe sur lesquels des contraintes sont spécifiées, et  $(Z_{C,i}, Z_{C,j}, \dots)$  sont les valeurs données par

l'utilisateur. Ces contraintes d'égalités sont écrites sous une forme matricielle, donnant :

$$M_{eq}Z = Z_{eq} \quad (4.8)$$

où  $M_{eq}$  est une matrice carrée remplie dont la valeur en diagonale est de 1 pour les points dont une contrainte de position est spécifiée et 0 pour les autres, et  $Z_{eq}$  correspond au vecteur colonne contenant les valeurs  $(Z_{C,i}, Z_{C,j}, \dots)$ .

#### 4.4.3 Ordonnement en profondeur des segments de courbe 2D

Maintenant que le dessin a été divisé en segments de courbes qui ont été assemblés en courbes simples fermées par l'utilisateur, il reste à définir comment les placer en 3D. Pour ce faire, on cherche à définir l'ordre de profondeur de ces courbes en leur attribuant un indice de profondeur  $I(C) \in \mathbb{D} = \{0, 0.5, 1, 1.5, 2, \dots\}$ . Si  $I(C_i) < I(C_j)$  alors cela signifie que la région définie par  $C_i$  est partiellement ou entièrement cachée par la région définie par  $C_j$  selon le point de vue utilisé pour le dessin. Ces indices de profondeur sont ensuite utilisés pour exprimer les contraintes d'inégalités auxquelles est soumis le problème d'optimisation, de manière à s'assurer que la profondeur relative des courbes 3D soit cohérente avec le dessin fourni en entrée.

Par défaut les indices de profondeur des courbes sont définis à 0 ou 1, en fonction de la visibilité de la courbe sur le dessin en entrée (section 4.2). Ces valeurs sont mises à jour en analysant les courbes 2D correspondantes et le type des jonctions à leurs points d'intersection dans le dessin. Nous avons déterminé 3 types de jonction qui doivent être pris en compte dans le calcul des indices de profondeur : les jonctions en T, les jonctions cuspidées et les jonctions en Y.

##### Jonction en T

La jonction en T est la plus simple : elle correspond à un croisement, une intersection, entre deux courbes, en un seul point. Ces jonctions vont toujours par deux, car elles apparaissent lorsque deux courbes fermées se chevauchent, et correspondent aux points où l'une de ces deux régions passe sous l'autre (fig. 4.13 (a)). Lorsqu'une paire est détectée le système regarde si au moins l'une des deux jonctions correspond à un point où une courbe passe de *visible* (avec un indice de 0) à *cachée* (avec un indice de 1 ou plus), le cas contraire les jonctions ne nous permettent pas de déterminer un ordre de profondeur entre ces deux éléments (fig. 4.4). Si l'on définit les deux courbes comme étant  $C_{2D,i}$  (l'élément au-dessus) et  $C_{2D,j}$  (l'élément passant en dessous) et leurs deux points d'intersection comme étant  $P_{T,1}$  et  $P_{T,2}$ , où  $P_{T,1}$  correspond au sommet où a lieu le changement de visibilité de  $C_{2D,j}$ , alors le système attribue à la partie de la courbe  $C_{2D,j}$  délimitée par  $[P_{T,1}, P_{T,2}]$  un indice de profondeur égal à  $n + 1$  avec  $n$  correspondant à l'indice de profondeur maximale de la courbe  $C_{2D,i}$  au niveau des deux jonctions. Cette règle de calcul des indices est utile lorsque de multiples éléments se superposent. Un exemple de calcul entre deux éléments d'un tel cas est représenté (fig. 4.13 (b)).

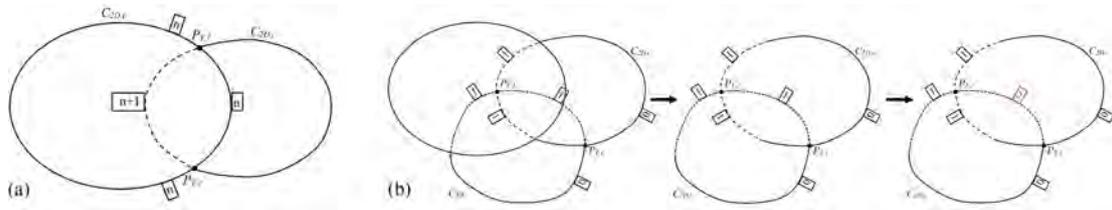


FIGURE 4.13 – Méthode de calcul des indices de profondeur lorsque deux courbes se chevauchent (a). Exemple où cette règle permet de déduire l’ordonnancement entre deux éléments appartenant à un chevauchement de 3 éléments (b). Initialement les traits pleins ont un indice de 0 et les pointillés de 1. Une fois que le programme détecte la paire de jonctions en T entre les deux éléments  $C_{2D,i}$  et  $C_{2D,j}$ , il met à jour la valeur de l’indice de profondeur du segment  $C_{2D,j}$  délimité par  $[P_{T,1}, P_{T,2}]$  comme étant la valeur maximale des indices de  $C_{2D,i}$  au niveau de ces jonctions plus 1, c’est-à-dire dans ce cas à 2.

### Jonction cuspidée

Tout comme pour les jonctions en T, les jonctions cuspidées correspondent à un point qui marque la fin de visibilité d’une courbe, et donc son passage en courbe cachée. Cependant, contrairement aux jonctions en T, les jonctions cuspidées ne correspondent pas à une intersection avec une autre courbe. Un point de jonction cuspidée  $P_C$  est un point où une courbe  $C_{3D,i}$  rentre à l’intérieur d’un volume délimité par un autre élément, c’est-à-dire qu’une partie de l’élément défini par  $C_{3D,i}$  est visible et que l’autre partie est cachée, car elle se situe à l’intérieur du volume 3D d’un autre élément fig. 4.14 (a).

Si l’on définit  $n$  comme étant l’indice de profondeur de  $C_{2D,i,1}$ , alors un indice de  $n + 0.5$  est assigné à sa partie cachée. Les indices ayant une partie décimale de 0.5 indiquent que la courbe 2D se situe à l’intérieur d’un autre élément en 3D, c’est-à-dire entre sa face avant et sa face arrière ; et permet ainsi de différencier des indices entiers qui indiquent qu’une courbe est complètement derrière ou devant un élément 3D, comme c’est le cas pour les jonctions en T.

### Jonction en Y

Une jonction en Y est un point où deux courbes se rejoignent pour se chevaucher, c’est-à-dire qu’elles traversent, à la suite de cette jonction, un même ensemble de points consécutifs. Les jonctions Y se présentent sous deux formes différentes : les jonctions Y sans segment cachés et les jonctions Y avec un segment caché. Un exemple avec des jonctions Y avec un segment caché est présent fig. 4.14 (b). Contrairement aux deux autres types de jonctions, celle-ci ne permet pas de déterminer les indices de profondeur, et n’est utilisé que pour nous indiquer que les éléments partagent des points en commun en 2D, et donc en 3D également.

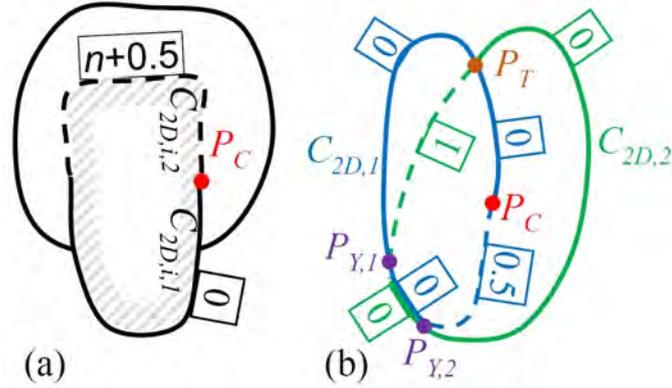


FIGURE 4.14 – Calcul des indices de profondeur aux courbes connectées à une jonction cuspidée au point  $P_C$  (a) ; les courbes cachées sont tracées en pointillés. Exemple d'un dessin composé de 2 courbes  $C_{2D,1}$  et  $C_{2D,2}$  avec une jonction en T au point  $P_T$ , une jonction cuspidée en  $P_C$  et deux jonctions en Y aux points  $P_{Y,1}$  et  $P_{Y,2}$  (b).

### Expression en contraintes d'inégalités

Afin de calculer les indices de profondeur de chaque segment de chaque courbe, nous affectons d'abord un indice de profondeur de 0 à tous les segments visibles dessinés en traits pleins, et de 1 aux autres. Nous mettons ensuite à jour les indices en analysant les jonctions des courbes et en appliquant les règles ci-dessus. Il reste cependant possible que l'indice de profondeur de certains segments ne puisse être déterminé correctement (voir fig. 4.4 (d)). Dans ce cas, le segment se voit attribuer le même indice de profondeur que son segment voisin. L'utilisateur peut également fournir manuellement la bonne valeur.

Nous devons par la suite utiliser ces indices de profondeurs pour écrire des contraintes d'inégalité pour chacun des segments de courbe. Le but de ces inégalités est de faire en sorte de conserver une distance minimale le long de l'axe  $z$  entre les courbes des éléments de surface qui se chevauchent dans le plan d'esquisse  $(x,y)$ , tout en conservant l'ordre de placement relatif des éléments les uns par rapport aux autres représenté dans le croquis.

Soit  $C_{3D,i,j}$  un segment de courbe et  $d_{i,j}$  son indice de profondeur. Si  $d_{i,j}$  est un chiffre entier, cela implique que ce segment est situé entre deux éléments surfaciques le long de l'axe  $z$ . Nous identifions alors  $E_B$  et  $E_F$ , qui sont les éléments derrière et devant  $C_{3D,i,j}$  respectivement, en regardant les segments voisins de  $C_{3D,i,j}$  et leur indice de profondeur (l'élément situé devant  $C_{3D,i,j}$  ayant un indice équivalent à  $d_{i,j} - 1$ , et l'élément derrière un indice à  $d_{i,j} + 1$ ). Nous devons alors écrire deux ensembles d'inégalités, une par rapport à  $E_B$ , l'autre par rapport à  $E_F$ .

Dans un premier temps nous écrivons les inégalités pour placer  $C_{3D,i,j}$  par rapport à  $E_B$ , ce qui revient à dire que l'on veut s'assurer que les coordonnées  $z$  de tous les points

de  $C_{3D,i,j}$  ont des valeurs plus élevées que celle de  $E_B$  :

$$M_{joncEB} \cdot Z > D_{joncEB} \quad (4.9)$$

où  $Z$  est le vecteur colonne de taille  $n$  contenant les coordonnées  $z$  de tous les points des courbes  $C_{3D}$ . Chaque ligne des matrices  $M_{joncEB}$  et  $D_{joncEB}$  correspond à une contrainte d'inégalité :

$$z_{p_K} > z_{p_{EB}}$$

où  $z_{p_K}$  est la coordonnée  $z$  d'un point  $p_K$  du segment de courbe  $C_{3D,i,j}$  et  $z_{p_{EB}}$  est la coordonnée  $z$  d'un point  $p_{EB}$  de la face avant de l'élément  $E_B$ . Comme l'on veut s'assurer que  $C_{3D,i,j}$  est devant  $E_B$ , il suffit de nous assurer que  $C_{3D,i,j}$  est devant la face avant  $E_B$ , c'est-à-dire que les coordonnées  $z$  de  $C_{3D,i,j}$  soient plus grandes que celle de la face avant de  $E_B$ . Notons que  $p_K$  est un point de la courbe  $C_{3D,i}$ , alors que  $p_{EB}$  n'est pas un point d'une courbe, mais est un point situé dans la région délimitée par la courbe de  $E_B$  ayant les mêmes coordonnées  $x,y$  que  $p_K$ .  $p_{EB}$  est exprimé sous la forme d'une somme pondérée des coordonnées  $z$  des points de la courbe de  $E_B$  (eq. (4.1), eq. (4.2) et eq. (4.3) dans section 4.3).

De la même façon, un ensemble d'inégalités est écrit pour placer  $C_{3D,i,j}$  par rapport à  $E_F$ , sauf que cette fois-ci  $C_{3D,i,j}$  doit être derrière  $E_F$ , ce qui revient à dire que les coordonnées  $z$  de tous les points de  $C_{3D,i,j}$  ont des valeurs plus faibles que celle de la face arrière de  $E_F$ . On écrit donc :

$$M_{joncEF} \cdot Z < D_{joncEF} \quad (4.10)$$

Chaque ligne de cet ensemble correspond donc à une inégalité entre un point  $p_l$  de  $C_{3D,i,j}$ , et son correspondant  $p_{EF}$  sur la face arrière de  $E_F$ . De cette manière chaque point est présent dans une paire d'inégalités visant à le placer entre la face avant de l'élément se trouvant derrière lui et la face arrière de l'élément se trouvant devant lui.

Si  $d_{i,j}$  est un chiffre avec une partie décimale de 0.5, ceci indique que  $C_{3D,i,j}$  est localisé à l'intérieur du volume d'un élément  $E_I$  (dans le cas d'une jonction cuspidée). Tout comme pour trouver  $E_F$  et  $E_B$  dans le cas d'un indice entier, l'élément  $E_I$  se trouve en regardant les segments voisins de  $C_{3D,i,j}$  et leurs indices de profondeurs. On écrit alors deux ensembles d'inégalités, sauf que cette fois-ci, ces ensembles sont écrits par rapport aux 2 faces du même élément  $E_I$  et non pas par rapport à deux éléments distincts. Nous avons donc un premier ensemble d'inégalité dont le but est de s'assurer que les coordonnées  $z$  des points de  $C_{3D,i,j}$  sont plus petit que ceux de la face avant de  $E_I$ , c'est-à-dire que  $C_{3D,i,j}$  est derrière la face avant de  $E_I$  :

$$M_{joncEIF} \cdot Z < D_{joncEIF} \quad (4.11)$$

Chaque ligne de cet ensemble étant une inégalité de la forme

$$z_{p_l} < z_{p_{EIF}}$$

où  $z_{p_l}$  est la coordonnée  $z$  d'un point  $p_l$  du segment  $C_{3D,i,j}$  et  $p_{EIF}$  est le point de même coordonnée  $x,y$  situé sur la face avant de  $E_{IF}$ .

De la même manière on écrit également un ensemble d'inégalités dont le but est de placer les valeurs  $C_{3D,i,j}$  devant la face arrière de  $E_I$ , et donc d'avoir des coordonnées  $z$  de  $C_{3D,i,j}$  plus grandes que ceux de la face arrière de  $E_I$  :

$$M_{joncEIB} \cdot Z > D_{joncEIB} \quad (4.12)$$

Comme pour la face avant, chaque ligne de cet ensemble est une inégalité de la forme

$$z_{p_l} > z_{p_{EIB}}$$

où  $z_{p_l}$  est la coordonnée  $z$  d'un point  $p_l$  du segment  $C_{3D,i,j}$  et  $p_{EIB}$  est le point de même coordonnée  $x,y$  situé sur la face arrière de  $E_{IB}$ .

De cette façon, chaque point  $p_l$  du segment  $C_{3D,i,j}$  sera donc soumis à deux inégalités, une le plaçant devant la face arrière de  $E_I$ , l'autre la plaçant derrière la face avant de  $E_I$ . Cela a donc pour conséquence que le segment  $C_{3D,i,j}$  sera placé comme voulu à l'intérieur de  $E_I$ . La fig. 4.15 montre le cas d'une jonction cuspidée et les inégalités qui en sont déduites.

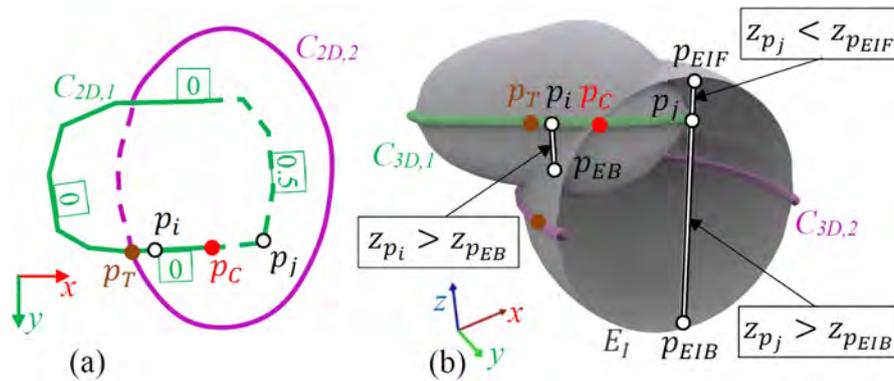


FIGURE 4.15 – Les courbes  $C_{2D,1}$  et  $C_{2D,2}$  avec une jonction en T au point  $P_T$  et une jonction cuspidée au point  $P_C$  (a). Les contraintes d'inégalité que l'on en déduit pour les points  $p_i$  et  $p_j$  appartenant à la courbe  $C_{3D,i}$  (b).

#### 4.4.4 Résoudre le problème d'optimisation

Une fois l'ensemble des jonctions analysé, nous obtenons un ensemble de contraintes d'inégalités visant à placer les éléments en profondeur les uns par rapport aux autres. Nous avons également un ensemble de contraintes d'égalités, les contraintes de position spécifiées par l'utilisateur. Nous nous retrouvons alors avec un problème d'optimisation linéaire que nous devons résoudre dans le but de déterminer les coordonnées  $z$  de tous

les points des courbes du dessin et où notre énergie de planarité est soumise à deux ensembles de contraintes, :

$$\min_Z \left\| \sum_{j=1}^m \alpha_j \cdot R_{C_{3D},j} \cdot Z \right\|^2 \text{ tel que } \begin{cases} M_{eq} \cdot Z = Z_{eq} \\ M_{ineq} \cdot Z < D_{ineq} \end{cases}$$

avec  $Z$  étant le vecteur contenant les coordonnées  $z$  de tous les points de  $C_{3D}$  :

$$Z = (z_1 \ z_2 \ \dots \ z_n)^T$$

L'égalité matricielle  $M_{eq} \cdot Z = Z_{eq}$  correspond à eq. (4.7), c'est-à-dire aux contraintes de position spécifiées par l'utilisateur. Et l'inégalité  $M_{ineq} \cdot Z < D_{ineq}$  est composée des inégalités obtenues par eq. (4.9), eq. (4.10), eq. (4.11) et eq. (4.12). Ce problème d'optimisation linéaire peut ne pas avoir de solution. C'est le cas lorsque les éléments représentés sur le dessin sont si étroitement entrelacés que les contraintes d'inégalité issues des jonctions en T ne peuvent pas être satisfaites toutes ensemble, comme pour un entrelacement complexe de tuyaux par exemple. Dans ce genre de cas, l'utilisateur pourra changer les indices de profondeur manuellement dans le but de modifier les contraintes d'inégalité afin de rendre le problème d'optimisation solvable.

Notez également qu'à l'inverse, cette optimisation peut avoir un nombre infini de solutions. Cela se produit lorsqu'aucune contrainte de position n'est définie par l'utilisateur, dans ce cas l'optimisation n'inclut aucune contrainte d'égalité, et peut être positionnée à n'importe quelle coordonnée  $z$  du moment que l'ordre relatif des éléments est correct.

Le problème d'optimisation est ensuite résolu par un solveur linéaire utilisant la méthode des moindres carrés. Parmi les nombreux algorithmes disponibles [NW06], nous avons choisi l'algorithme des points intérieurs [Gon95]. Nous avons fait ce choix, car celui-ci était le plus efficace, et permet de trouver une solution pour les cas les plus complexes en un maximum de 2000 itérations. Un autre avantage de cet algorithme est que, même lorsque l'optimisation comporte un nombre infini de solutions, celui-ci va néanmoins calculer et retourner une solution.

La solution de cette optimisation est alors le vecteur  $Z$  contenant l'ensemble des coordonnées  $z$  des points des courbes, et permettra de placer celles-ci dans l'espace 3D. Cet ensemble de courbes 3D est ensuite présenté dans une visionneuse 3D, permettant à l'utilisateur de voir le résultat sous tous les angles. Il peut ensuite, s'il le désire, ajouter de nouvelles contraintes de position ou de nouveaux points d'angle afin d'affiner le résultat en calculant une nouvelle optimisation. Ce processus de modification peut être répété jusqu'à l'obtention de la forme 3D souhaitée.

## 4.5 Étude utilisateur

Dans le but de valider la robustesse de notre méthode, ainsi que sa convivialité et sa facilité de prise en main, nous avons conduit une étude utilisateur où il a été demandé

aux participants d'utiliser Descriptive pour produire des modèles 3D correspondant à des modèles référant qu'on leur a montré, puis d'évaluer l'efficacité et l'intuitivité de notre système.

Huit participants ont accepté de prendre part à notre étude, 7 hommes et 1 femme. De manière à ne pas biaiser nos résultats par de quelconques connaissances a priori sur le sujet, ou la modélisation géométrique 3D de manière générale, nous avons pris garde à avoir une répartition équilibrée d'utilisateurs avec et sans connaissance. Ainsi, parmi nos sujets quatre d'entre eux sont des étudiants diplômés en informatique ayant soit une formation en infographie (P1, P2 et P3 dans les graphes suivants) soit en web design (P8). Les quatre autres sont des étudiants diplômés d'autres domaines scientifiques, tels que la physique, la chimie ou les mathématiques.

Le second point important est que certains de ces utilisateurs (P2, P5 et P8) ont déjà de l'expérience avec des logiciels de modélisation 3D tels que Blender ou 3DS Max, mais que les autres sont tout à fait novices. C'est cette différence de niveaux qui nous permettra de tester l'intuitivité et l'efficacité du système aux yeux d'un utilisateur non-expert, tout en ayant un comparatif possible avec les logiciels 3D par les habitués.

Chacun de ces participants a suivi la même procédure d'évaluation, individuellement. Cette procédure est décomposée en 3 étapes, et suit un déroulement classique : (1) l'utilisateur passe d'abord par une session d'entraînement (2), il effectue ensuite, seul, les modélisations requises et (3) répond ensuite à un questionnaire nous permettant de récolter leurs retours et impressions.

**Session d'entraînement** Cette session est conçue en deux temps dans le but d'enseigner à l'utilisateur comment utiliser Descriptive dans son ensemble. Dans un premier temps, il lui est expliqué en quoi consiste un dessin descriptif, comment guider la reconstruction à l'aide des spécifieurs et comment utiliser l'interface utilisateur de notre modèleur. Dans un second temps, chaque utilisateur a été guidé par un instructeur afin de comprendre le processus complet de modélisation. Trois modèles référents, ainsi que toutes les étapes nécessaires à leur reconstruction, en commençant par le dessin, leur ont été présentés en détail. Au cours de cette étape, les participants ont été autorisés à tester eux-mêmes comment modifier le résultat en modifiant / ajoutant / supprimant des spécifieurs locaux, et en demandant à l'instructeur s'ils avaient besoin d'informations supplémentaires. Les exemples d'apprentissage sont joints à l'appendix B.

La durée de cette session varie entre 12 et 35 minutes, en fonction des participants, et a été en moyenne de 21 minutes. Notons que cette durée est bien moins importante que le temps nécessaire pour qu'un utilisateur apprenne à utiliser un logiciel de modélisation classique tel que 3DS Max, dont le temps de prise en main est de plusieurs heures minimum voire plusieurs jours. Le fait de permettre à n'importe quel utilisateur d'avoir accès à la reconstruction 3D en ne nécessitant qu'un temps d'entraînement très faible correspond à ce que l'on attend des modèleurs à base de croquis, dont le but premier est de rendre la modélisation géométrique 3D accessible à tous. Ainsi avec un temps moyen

de prise en main de 21 minutes, Descriptive se montre être un outil intuitif et facile d'accès.

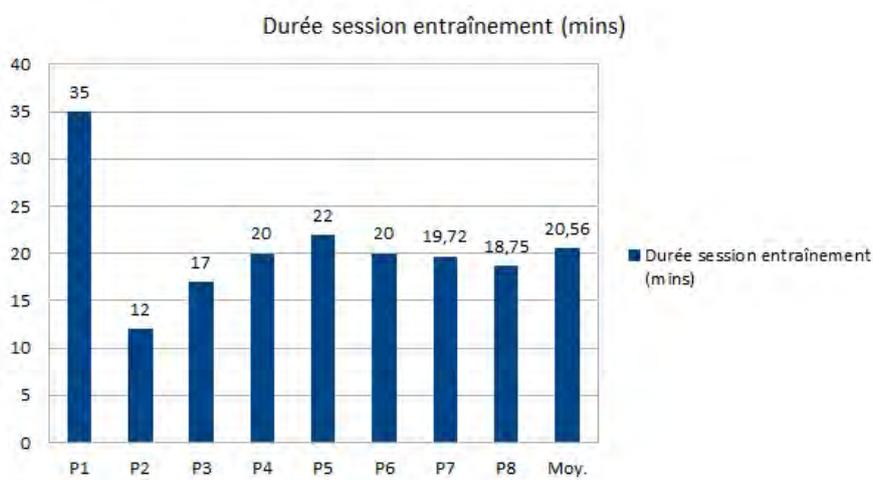


FIGURE 4.16 – Durée de la session d'entraînement pour chaque participant.

**Session de modélisation** Une fois que les participants ont compris comment utiliser Descriptive, ils sont invités à modéliser trois modèles spécifiques. L'instructeur n'est alors présent que pour leur montrer le modèle référent et répondre à leurs questions, uniquement si celles-ci portent sur l'interface. Pour chaque reconstruction, il est d'abord demandé à l'utilisateur de fournir un dessin descriptif correspondant au modèle référent qui lui est présenté. Cette étape entraîne inévitablement une différence notable entre le modèle reconstruit et le modèle référent, du à la non-exactitude entre le dessin fourni par les participants et les contours du modèle référent (différence inhérente à la capacité artistique de chacun). Chaque participant effectue les trois reconstructions dans un ordre fixe : fenêtre, pantalon et T-rex.

Le temps nécessaire à la modélisation géométrique varie évidemment en fonction du modèle référent à reconstruire et du participant, mais les résultats de notre étude nous montrent que la durée moyenne requise, pour tout le processus de reconstruction, se situe entre 6 et 8 minutes, y compris pour des utilisateurs novices. Ce temps moyen valide le fait que Descriptive permette à n'importe quel utilisateur d'obtenir rapidement une reconstruction 3D. La fig. 4.17 montre le meilleur résultat pour chaque modèle référent, ainsi que les dessins et spécificateurs utilisés pour leur obtention ; l'intégralité des résultats de cette étape de modélisation est disponible appendix C. Par ailleurs la fig. 4.18 présente le temps requis par chaque participant pour effectuer la reconstruction de chacun des modèles référents.

Les résultats nous montrent également que pour certains, le dessin en style descriptif n'était pas pleinement compris pour la première reconstruction. La fenêtre étant leur

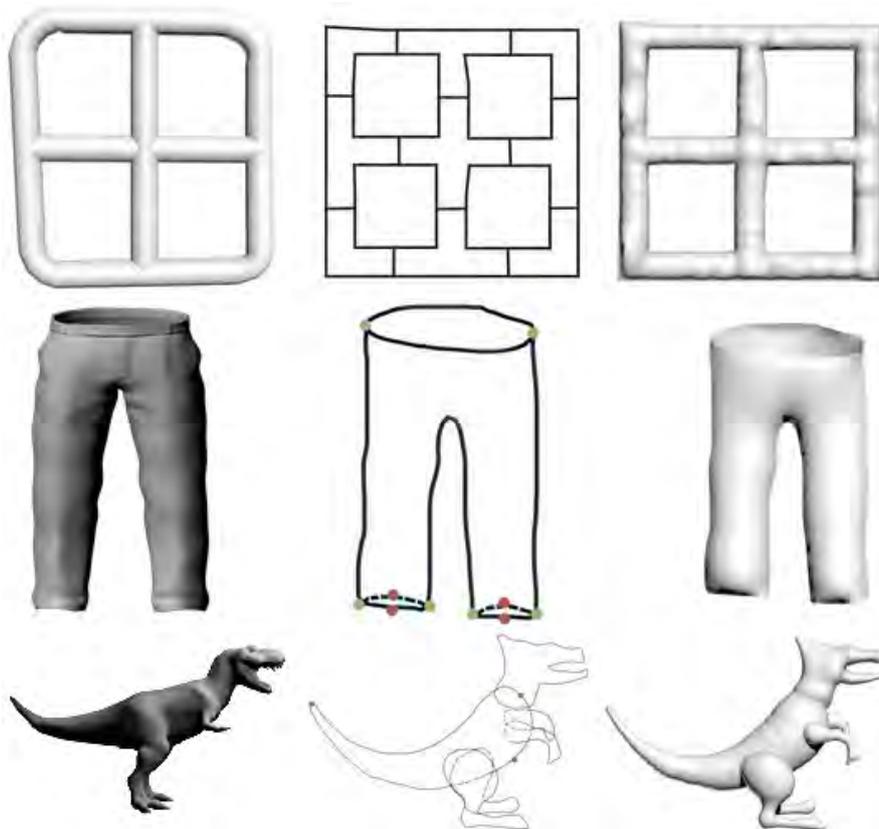


FIGURE 4.17 – Le meilleur résultat obtenu par les utilisateurs. De gauche à droite : les modèles référents, l’esquisse fournie par l’utilisateur avec les points d’angle (en vert) et les contraintes de position (en rouge), ainsi que les modèles reconstruits.

premier modèle à reconstruire intégralement, l’obligation de chevaucher les différentes parties de l’objet pour obtenir des jonctions lisses n’était pas évidente pour tout le monde ni pour chaque partie. (P2 par exemple recouvre correctement le contour de la fenêtre, mais pas l’intérieur). Ce n’est bien sûr pas le cas pour tous, par exemple P8 a obtenu un résultat parfait ; de plus ces confusions sur la façon de dessiner ont tendance à disparaître à mesure que les participants utilisent Descriptive.

**Questionnaire post-reconstruction** Après l’étape de modélisation, les commentaires de chaque participant ont été collectés par le biais de quatre questions. Le but de ce questionnaire étant de récolter leurs ressentis sur Descriptive et son utilisation, nous leur avons demandé d’accorder à chacune d’entre elles un score allant de 1, pour très négatif, à 5, pour très positif. Ils ont donc dû répondre aux questions suivantes :

1. Est-il facile de dessiner avec le style descriptif ?

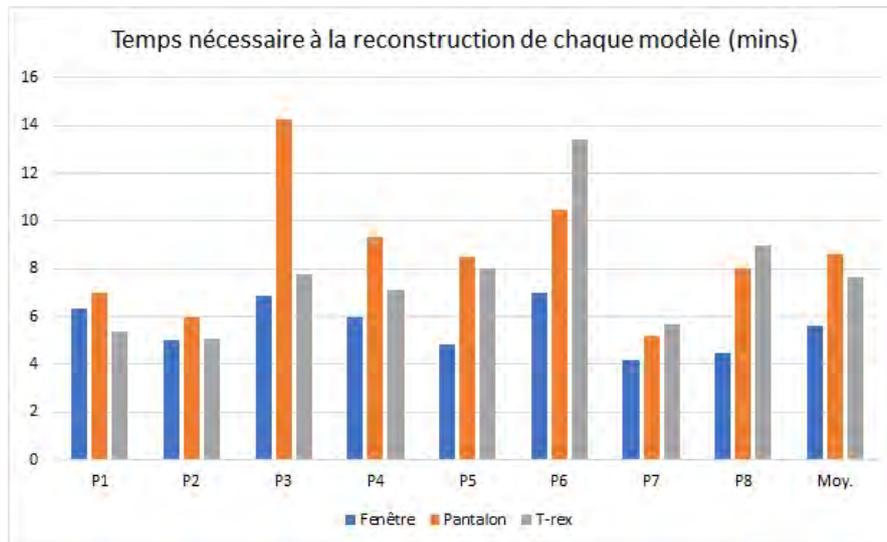


FIGURE 4.18 – Temps requis pour chaque participant pour effectuer la reconstruction du modèle 3D référent correspondant.

2. Lorsque le premier résultat 3D ne correspondait pas au résultat attendu, était-il facile de le corriger ?
3. Le modeleur, dans sa globalité, est-il facile à utiliser ?
4. Au vu du dessin que vous avez fourni en entrée, le résultat 3D final correspond-il à vos attentes ?

En moyenne, les utilisateurs ont attribué une note de 4,6 à la facilité de dessin en style descriptif et une note de 4,1 à la facilité d'utilisation globale de notre modélisateur. Ils ont également attribué une note de 4,75 à la correspondance entre leurs résultats 3D et leurs attentes. Ces résultats nous montrent que les participants ont trouvé le système intuitif, bien qu'encore perfectible, et que la qualité des résultats fournis par Descriptive est plus que satisfaisante. Le détail de leurs réponses se trouve dans le graphe fig. 4.19.

De plus, il a été demandé aux utilisateurs ayant une expérience antérieure avec des logiciels de modélisation 3D de répondre à trois questions supplémentaires, permettant ainsi d'obtenir leur avis sur la comparaison d'efficacité et de facilité entre Descriptive et leur logiciel de modélisation habituel.

1. Descriptive est-il plus facile à prendre en main, et donc à utiliser, que les outils que vous utilisez couramment ?
2. Descriptive vous a-t-il permis d'obtenir un résultat 3D plus rapidement que si vous aviez utilisé un de vos outils habituels ?
3. Aimez-vous ce principe de modélisation basée sur l'esquisse de traits 2D ?

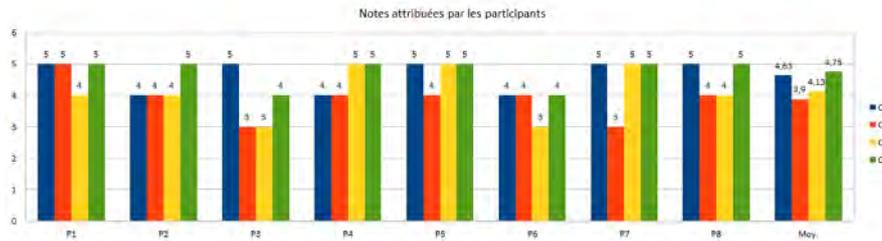


FIGURE 4.19 – Note attribuée par les participants aux questions de notre évaluation après avoir effectué leurs reconstructions.

Les trois participants ont été unanimes sur le fait qu'un procédé de modélisation à partir de croquis 2D était vraiment appréciable, et qu'il était possible d'obtenir un résultat relativement plus rapidement qu'avec leurs outils de modélisation 3D habituels, notamment au vu de la facilité de prise en main du système. Le détail de leur réponse est présent dans le graphique fig. 4.20 ci-dessous.

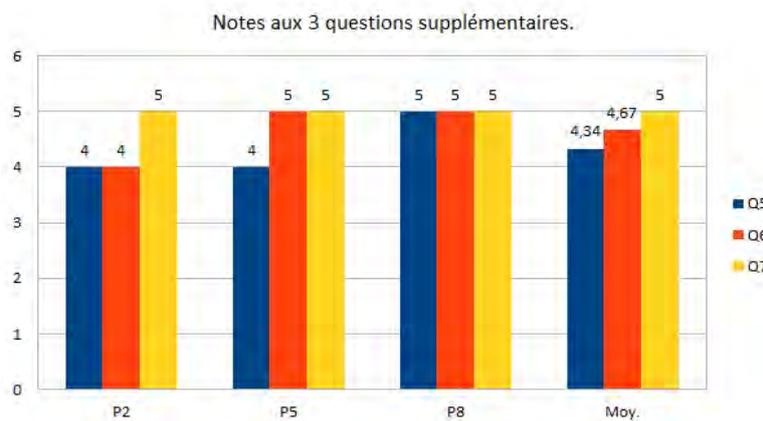


FIGURE 4.20 – Note attribuée par les participants avec une expérience antérieure avec des logiciels de modélisation 3D aux trois questions supplémentaires.

En résumé, cette étude utilisateur a permis de montrer qu'après une formation d'une durée moyenne de 21 minutes, durée relativement courte si l'on compare avec celle nécessaire pour l'utilisation de modélisateurs 3D classiques, les utilisateurs sont capables de créer leurs propres formes à l'aide de Descriptive, de manière rapide (6 à 8 minutes en moyenne) et en obtenant des résultats 3D qui les satisfont. Descriptive remplit donc des critères qu'un modélisateur à base de croquis se doit d'avoir : être facile à prendre en main pour tous, sans nécessiter de connaissance a priori sur la modélisation 3D, et permettre une reconstruction rapide de modèles de qualité.

## 4.6 Résultats et discussion

Nous avons décidé d'implémenter Descriptive en C++ en tant que plug-in pour 3DS Max fonctionnant sous Windows 10. Ce choix a été effectué pour plusieurs raisons. Dans un premier temps, cela permet d'avoir directement accès aux fonctionnalités de base de 3DS Max qui sont communes avec notre objectif : une visionneuse 3D ainsi que des outils de sélection dans cette vue. Dans un second temps, 3DS Max est un logiciel de modélisation couramment utilisé par les habitués, et sortir Descriptive sous forme de plug-in nous permet d'étendre les capacités du logiciel, et d'offrir plus facilement accès à ces extensions à de nombreux artistes.

Un ensemble de résultats produits avec Descriptive est présenté fig. 4.21. Les contraintes spécifiées par l'utilisateur sont représentées en rouge pour les contraintes de positions, en vert pour les points d'angle et vert-rouge pour les deux à la fois. Ces résultats ont été obtenus en utilisant le plug-in sur une station de travail dotée de 16 Go de mémoire et d'un processeur Intel Core i7-6700 fonctionnant à 3,4 GHz.

Comme on peut le voir, Descriptive est capable de modéliser une grande variété de formes, que ce soit des formes organiques, telles que le dinosaure fig. 4.21(a) par exemple, ou des objets artificiels, tels qu'un avion fig. 4.21(h) ou une chaise fig. 4.21(b). De plus, bien que Descriptive se base sur des courbes fermées et la reconstruction d'éléments surfaciques fermés, il est tout de même possible de créer des surfaces qui semblent ouvertes. Grâce à la possibilité de définir une épaisseur négative pour un côté d'un élément, et positive pour l'autre (section 4.3), Descriptive est capable de reconstruire des objets concaves tels qu'un vase fig. 4.21(f) ou un tuyau fig. 4.21 (i). Ce dernier modèle montre également qu'il est possible de réaliser des formes à multiples couches. En effet, à l'intérieur du tuyau se trouve un élément 3D représentant un liquide. Ceci se voit également avec l'ourson en peluche fig. 4.21(c), qui a été modélisé avec une chemise sur lui. Ces éléments englobants (tuyau et chemise) sont constitués de deux éléments de surface concaves, chacun représentant la partie avant et arrière de l'élément englobant. De plus, l'utilisateur peut également spécifier une épaisseur nulle aux éléments, permettant ainsi la reconstruction d'objets entièrement polyédriques, comme une maison fig. 4.21 (j), composée uniquement de surfaces planes. Ces éléments plats sont obtenus par  $\beta V_{fr,j} = \beta V_{bk,j} = 0$  dans les eq. (4.2) et eq. (4.3).

Un autre point important montré par ces résultats est l'efficacité avec laquelle notre méthode gère la reconstruction des parties cachées dans les objets, et comment elle arrive à reconstruire sans soucis des objets contenant beaucoup d'auto-occultation et d'éléments qui s'entrelacent, tels que la chaise fig. 4.21 (b) et les branches sortant du verre fig. 4.21 (d). Descriptive est également capable de trouver une reconstruction même lorsque la position relative des éléments entre eux forme un cycle, ce genre de forme étant souvent une limitation des autres méthodes. La fig. 4.21 (e) montre un tel cas, chaque disque est

au-dessus du suivant et en dessous du précédent, formant ainsi un cycle où chacun doit être au-dessus de son voisin.

Descriptive offre également une liberté sur l'angle de vue de l'objet représenté sur le dessin. Comme on peut le voir, nos résultats ont été faits sous plusieurs directions différentes : de face, de profil, de trois quarts. Il est également possible de reconstruire un modèle avec des orientations différentes pour chaque partie de l'objet, tel que le kangourou de la fig. 4.21 (h) où le bas du corps est de profil, mais la partie haute, et principalement la tête, est de trois quarts.

Ces résultats montrent également que le nombre de contraintes nécessaires pour la reconstruction de chaque modèle est minime. Pour le dinosaure par exemple, seules trois contraintes de position ont été nécessaires, placées sur des points arbitrairement choisis et en fixant une valeur à  $Z=0$  avec pour seul objectif de placer le modèle reconstruit sur le plan  $Z=0$ . Sans ces 3 contraintes, il existe une infinité de reconstructions du même modèle 3D, mais sur des plans  $Z$  différents. Ce résultat 3D est donc en réalité obtenu directement à partir de la vue unique du dessin, sans avoir besoin de passer par des rotations d'autres vues en 3D. Pour les autres modèles, les contraintes de position sont principalement utilisées après la première reconstruction et servent à spécifier une position spécifique, parmi toutes celles possibles, correspondant à ce que l'artiste veut. Pour la chaise (b) par exemple l'utilisateur spécifie la position du haut et du bas de chaque pied de chaise de manière à s'assurer qu'ils soient tous parfaitement verticaux. Ainsi la spécification des contraintes est une tâche aisée, qui permet à l'utilisateur de modifier rapidement les résultats pour que ceux-ci correspondent à leurs attentes.

Enfin, le temps requis pour produire ces résultats, pour l'intégralité du processus (temps de dessin compris), varie de 5 minutes (hélice de la fig. 4.21 (e)) à 25 minutes (arbre enchevêtré de la fig. 4.21 (d)). Il a fallu en moyenne 27 secondes pour détecter les jonctions, attribuer les indices de profondeur et exprimer le problème d'optimisation avec les contraintes, et 20 secondes pour le résoudre. Notez que le temps de calcul dépend grandement de la complexité de la forme (généralement le nombre d'éléments fermés et de jonctions) présente dans l'esquisse. Pour l'arbre enchevêtré (fig. 4.21 (d)), ces temps sont respectivement de 54 et 41 secondes pour un problème d'optimisation avec un vecteur  $Z$  de 1073 variables et soumis à 766 contraintes, tandis que pour l'hélice (fig. 4.21(e)), ils sont presque instantanés (1 et 2 secondes, respectivement, pour uniquement 103 variables et 25 contraintes). Descriptive permet donc de produire une large variété de formes, dans des temps relativement faibles. D'autres formes produites avec notre modèleur sont présentes dans l'appendix D.

#### 4.6.1 Comparaison avec les travaux précédents

Cette section a pour but d'effectuer une comparaison avec d'autres méthodes de reconstruction à l'aide d'esquisses présentées section 2.9. Notre objectif est de pointer

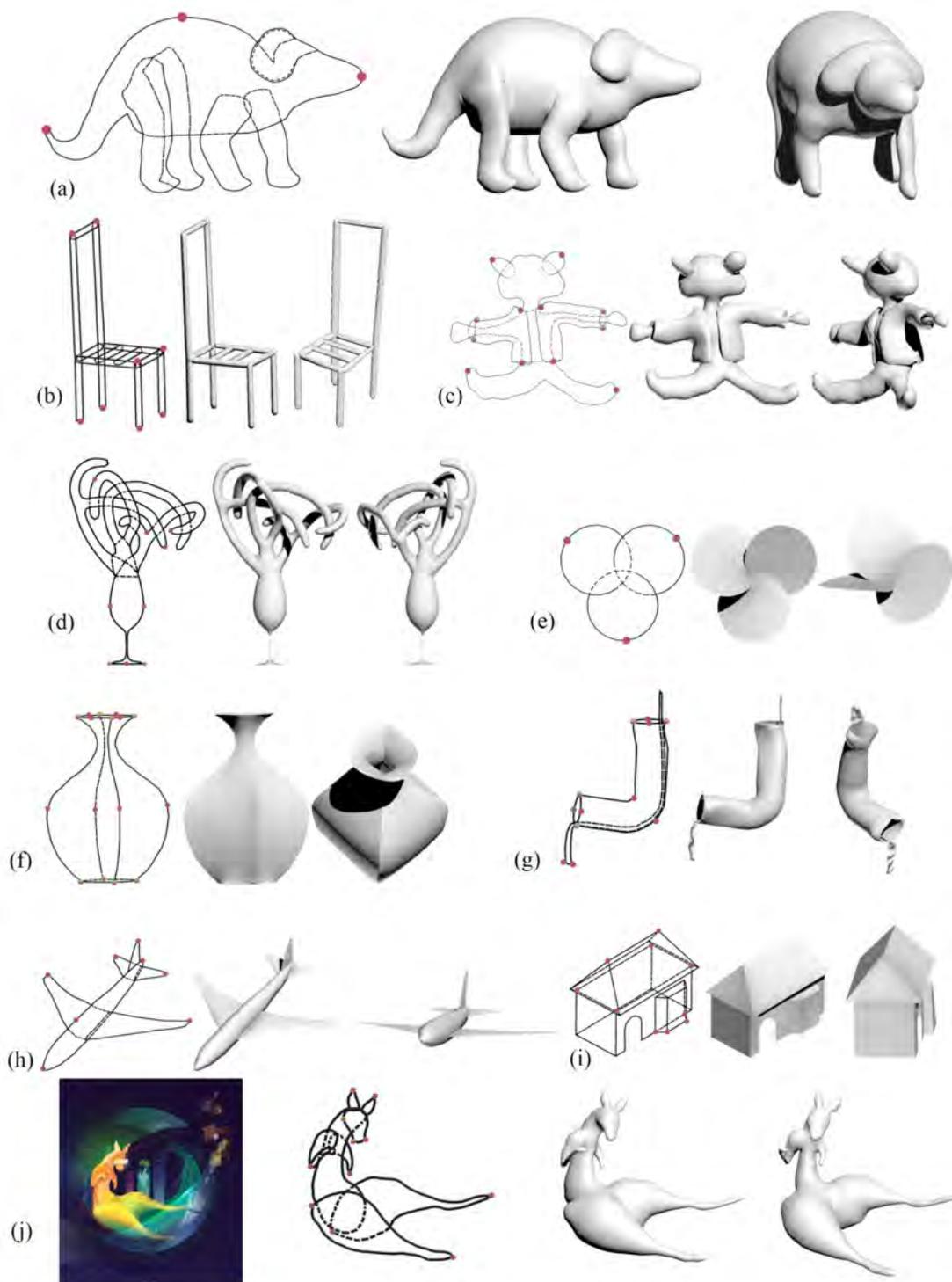


FIGURE 4.21 – Ensemble de résultats obtenus avec notre système. Les contraintes spécifiées par l'utilisateur sont représentées en rouge pour les contraintes de position, en vert pour les points d'angle et vert-rouge pour les deux à la fois.

les formes qui peuvent être reconstruites avec les autres méthodes, nous cherchons donc à savoir si celles-ci peuvent modéliser l’objet en supposant que leurs données d’entrées correspondent à celles requises par leur méthode. On considère donc que les données fournies en entrée de chaque méthode sont conformes aux spécifications de cette méthode et non à celles de Descriptive. La fig. 4.22 présente donc un tableau mettant en avant quels résultats parmi ceux que Descriptive est capable de produire, fig. 4.21, pourraient être reconstruits si l’utilisateur choisissait d’utiliser une autre méthode existante. Pour faire une comparaison la plus objective possible, celle-ci est effectuée avec un ensemble de méthodes choisies pour représenter tous les types de reconstruction différents. Nous avons donc *BendSketch* de [LPL<sup>+</sup>17] pour les méthodes utilisant de multiples vues, *SecondSkin* de [PS15] et *Urban Procedural Model* de [WMW<sup>+</sup>08] pour les méthodes guidées par des données 3D et des a priori, *Ink-and-ray* de [SKv<sup>+</sup>14] pour les méthodes de bas-relief, *Robust Flow-Guided Neural Predictions* de [LPL<sup>+</sup>18] pour les réseaux de neurones, *True2Form* de [XCS<sup>+</sup>14] pour les méthodes basées sur les réseaux de courbes spécifiques aux objets artificiels et *Automatic Structuration of organic shapes* de [EPB<sup>+</sup>19] pour les objets organiques. Nous avons également ajouté *Teddy* le précurseur de toutes les méthodes, de [IIMT07] ainsi que *Sketching of self-occluded object* de [CS07] qui utilisent également des indices de profondeur pour déterminer un placement relatif des éléments.

	a	b	c	d	e	f	g	h	i	j
BendSketch	X	X			X	X		X		X
SecondSkin			X							
Sketching of self occluded object				X						
Ink-And-Ray	X									X
Robust Flow-Guided Neural Prediction					X	X		X		
Teddy	X	X						X		X
Urban procedural model									X	
True2Form		X			X	X	X	X		
Automatic structuration of organic shapes	X			X						X

FIGURE 4.22 – Comparaison avec les autres méthodes. Les cases sont cochées quand la méthode est capable de reconstruire le modèle correspondant, en supposant qu’elle ait les données adéquates en entrée.

Les méthodes comparées peuvent être grossièrement classées dans deux catégories différentes : les méthodes limitées à une classe d’objets, et les méthodes permettant la reconstruction de formes libres.

**Méthodes limitées à des types spécifiques.** Certaines méthodes sont conçues pour ne fonctionner qu’avec un type de forme spécifique et ne peuvent donc reconstruire que ce type d’objets. Par exemple *Automatic structuration of organic shapes* est développé pour reconstruire des formes organiques, et pourra donc reconstruire des formes telles que le dinosaure (a), le kangourou (j) et sûrement l’arbre (d), mais ne pourra pas reconstruire un objet fabriqué par l’homme tel que l’avion (h) ou la maison (i). Cette méthode ne gère pas non plus la reconstruction de formes présentant un cycle, tel que notre hélice (e). À l’inverse, certaines méthodes, telles que *True2Form*, sont conçues pour gérer les modèles artificiels et peuvent également produire un tuyau (g), mais ne sont pas capables de reconstruire les formes organiques mentionnées précédemment.

Les méthodes guidées par des données ou des a priori sont également restreintes à des types de reconstruction spécifiques. De la même manière que notre méthode pour la reconstruction des fleurs est limitée aux objets floraux, *Urban Procedurale Model* ne peut reconstruire que des bâtiments. *SecondSkin* a besoin d’un modèle 3D qui sert de base pour reconstruire d’autres éléments 3D. Il peut donc produire un résultat pour chaque forme pouvant être décomposée en couches que nous placerions sur la géométrie 3D sous-jacente. Cette méthode pourrait produire le tissu de l’ours en peluche (c) si nous acceptons qu’ils aient l’ours comme géométrie 3D sous-jacente. En plus de ne produire qu’un maillage en bas-relief, *Ink-and-Ray* rencontre des difficultés avec les objets polyédriques tels que la maison et ne peut pas gérer la forme avec un trou complet (vases (e, f), tuyau (g)).

**Méthode pour des formes libres.** Les autres méthodes avec lesquels nous effectuons la comparaison sont conçues pour produire un résultat 3D de formes libres, donc non restreint à un type particulier. Cependant on remarque qu’elles ne permettent pas la même diversité de construction que notre modélisateur. *Teddy* est capable de reconstituer des formes organiques (dinosaure (a), kangourou (b)) ainsi que des objets fabriqués par l’homme simple (chaise (b), avion (h)); mais ne peut pas produire un résultat qui nécessite des éléments parfaitement plats (maison (i), vases (e, f)) et cela devient difficile lorsque le dessin contient des éléments qui s’entrelacent énormément (arbre (d)) ou des éléments englobants comme la chemise autour de l’ours (c) ou le tuyau (g) dans lequel l’eau s’écoule. Le réseau de neurones *Robust Flow-Guided Neural Predictions* peut produire un résultat pour des esquisses qui peuvent être facilement décomposé en un nombre limité de surfaces sans occultation, telles que les vases (e, f) ou l’avion (h), mais ne peut pas gérer les formes plus complexes comme l’arbre (d) ou la chaise (b). De même que pour *Teddy*, il ne peut pas non plus gérer les éléments englobants. *BendSketch* peut produire un grand nombre de nos résultats, car l’utilisateur a la possibilité de tourner dans l’espace 3D et dessine étape par étape permettant ainsi de s’assurer de la forme obtenue. Cependant certains efforts supplémentaires seront nécessaires pour l’utilisateur pour quelques modèles, car celui-ci pourrait devoir modéliser certaines parties d’une forme séparément, et les assembler manuellement par la suite. De plus, l’arbre reste un modèle présentant beaucoup trop d’entrelacements pour que cette méthode puisse le gérer.

En conclusion, comparé aux méthodes déjà existantes, notre système permet la reconstruction d'un véritable maillage 3D (pas d'un relief tel que [SKv<sup>+</sup>14],[DNJ<sup>+</sup>18],[YHJ<sup>+</sup>17]) sans être limité sur la nature de l'objet reconstituable (que se soit un objet artificiel ou organique) ou à des objets possédant une géométrie particulière (telle qu'une symétrie [CSPyN11]). En permettant à l'utilisateur de modifier facilement la position de certains éléments, par le biais des contraintes de positions, les modèles reconstruits ne sont pas limités dans la profondeur et ne donnent pas cette impression de forme aplatie que peuvent avoir les modèles obtenus avec d'autres méthodes telles que [CS07] et [KH06]. Faire le modèle de la chaise avec ces méthodes serait impossible par exemple. De plus, à notre connaissance, très peu de modeleurs basés sur des croquis, y compris ceux qui permettent de reconstruire de manière incrémentale des formes très complexes telles que ShapeShop ([SWSJ07]), sont capables de faire la reconstruction à la fois d'un objet et d'un élément fin l'englobant (tel que le cylindre dans lequel coule de l'eau ou les vêtements autour de l'ourson en peluche). Comme le montre le tableau, il est effectivement possible de reconstruire chacun des modèles 3D présents dans nos résultats en utilisant pour chacun d'entre eux un modeleur spécifique adapté, mais Descriptive est capable à lui seul de reconstruire l'ensemble de ces résultats, et tout cela en ne nécessitant qu'un court temps d'apprentissage d'environ 20 minutes comme l'a montré notre étude utilisateur.

#### 4.6.2 Limitations

Bien que Descriptive soit une méthode qui permette de reconstruire rapidement et facilement une grande variété de modèles 3D, des améliorations sont encore possibles. Certains modèles 3D peuvent parfois nécessiter un peu trop de temps et d'efforts pour la mise en place des contraintes géométriques. C'est notamment le cas des formes dont les courbes de silhouette 3D comportent de très nombreux entrelacements. Une autre limitation de notre méthode est que la façon de dessiner les parties cachées, ainsi que la façon de regrouper les segments de courbes en éléments fermés, n'est pas toujours évidente pour un utilisateur non expérimenté. Comme le montre la fig. 4.23, un utilisateur peut se trouver dans une situation où plusieurs esquisses 2D apparemment correctes aboutissent à des modèles 3D complètement différents. Pour obtenir le modèle attendu, l'utilisateur doit donc avoir parfaitement saisi la façon dont le modeleur effectue la reconstruction, de manière à bien faire la différence entre ces similarités et donc faire le bon choix de représentation.

### 4.7 Conclusion

Ce chapitre a présenté le second travail réalisé au cours de ma thèse, Descriptive, un modeleur basé sur les croquis qui reconstruit les formes 3D à partir d'un seul dessin descriptif 2D. La motivation principale derrière ce travail était de développer une méthode qui ne souffrirait pas du même problème que notre travail pour la reconstruction 3D de fleurs, c'est-à-dire ne pas être limitée à un type d'objet particulier, tout en conservant

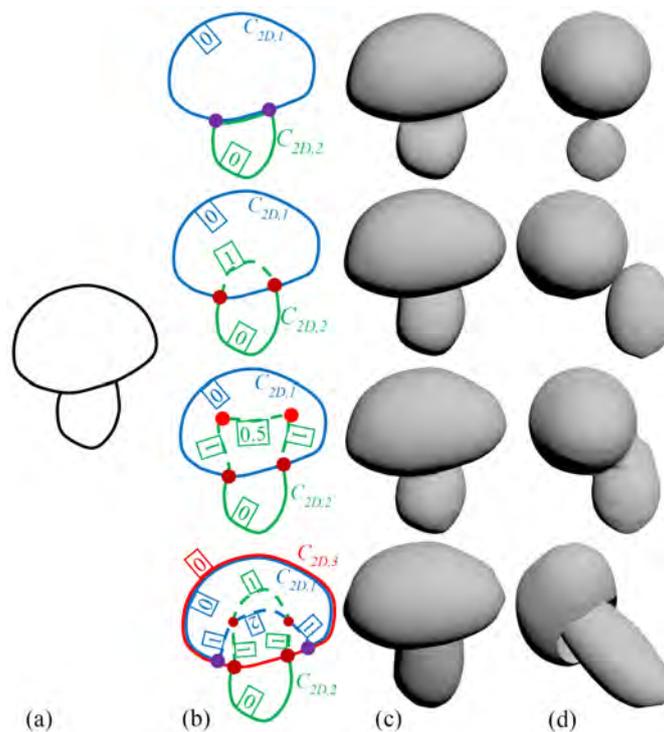


FIGURE 4.23 – La même esquisse (a) peut donner différentes formes en fonction de la manière dont l'utilisateur définit les courbes masquées et leur indice de profondeur (b). Les jonctions en T, les jonctions cuspidées et les jonctions en Y sont représentées en brun, rouge et violet respectivement. Les résultats de la reconstruction sont présentés en vue de face (c) et en perspective (d). La décomposition peut ne pas toujours être simple et peut nécessiter des informations sur le modélisateur.

les aspects clés de celle-ci : être facile d'utilisation, ne nécessiter qu'un seul dessin en entrée, et ne pas être restreint à un angle de vue en particulier.

En partant toujours du constat qu'un dessin unique classique ne fournit pas assez d'informations, nous avons décidé de développer un système qui prendrait en entrée un style un peu plus informatif. Cette idée était déjà présente dans des travaux existants, tels que [XCS<sup>+</sup>14], mais le style graphique choisi par ceux-ci était spécialement conçu pour l'industrie et les objets manufacturés, limitant ainsi leurs méthodes à la reconstruction de tels objets. C'est pourquoi nous avons choisi d'utiliser un style de dessin descriptif simple d'accès et intuitif pour tous, mais qui apporte malgré tout suffisamment d'informations sur les parties cachées pour permettre une reconstruction.

Il fallait également que notre système laisse à l'utilisateur un certain contrôle sur la reconstruction 3D de manière à ce qu'il soit capable de la modifier si elle ne lui convenait pas. En implémentant le système de spécificateurs, nous lui avons donné accès à des outils

simples d'utilisation, mais qui permettent un contrôle total sur la forme reconstruite. Ces spécifieurs permettent également à l'utilisateur de corriger les ambiguïtés présentes sur le dessin, ambiguïté que le système ne peut résoudre seul. En étant guidé par l'utilisateur, notre modeleur est donc extrêmement robuste.

Nous avons donc mis au point un système qui, en combinant les informations présentes dans un dessin en style descriptif avec la possibilité pour l'utilisateur de guider la reconstruction à l'aide de spécifieurs, élargit la gamme et la complexité des formes pouvant être créées à partir d'une esquisse à vue unique, tout en maintenant l'interface d'esquisse intuitive et conviviale. En exprimant le problème de reconstruction 3D en tant que problème d'optimisation linéaire avec des contraintes d'égalité et d'inégalité, notre modeleur est capable de produire un résultat dans des temps limités. En permettant la reconstruction rapide de formes sans contraintes sur le type d'objet ou l'angle de vue du dessin, en ne nécessitant qu'une seule esquisse en entrée et en étant accessible à tout type d'utilisateur, Descriptive remplit les critères qui ont motivé notre travail.

# Chapitre 5

## Conclusion

Les travaux effectués au cours de cette thèse visent à apporter de nouvelles solutions au problème de reconstruction géométrique à partir de croquis. L'un des objectifs principaux était que nos méthodes gardent la simplicité inhérente au dessin, c'est pourquoi elles partagent un principe commun : seul un croquis simple est fourni en entrée par l'utilisateur. Cependant, cela signifie qu'il est nécessaire d'obtenir les informations requises pour effectuer la reconstruction par d'autres moyens. Une discussion rapide sur les deux méthodes développées au cours de cette thèse est présentée section 5.1, puis la section 5.2 propose une perspective de recherche future.

### 5.1 Discussion

La première méthode que nous avons développée appartient à la catégorie des méthodes basées sur des a priori. En nous limitant à la reconstruction d'un type d'objets spécifique, les fleurs, nous avons réussi à mettre au point une solution permettant de produire rapidement des modèles floraux 3D d'une large variété d'espèces botaniques. Pour arriver à ce résultat, notre méthode se sert d'un ensemble d'a priori sur la structure de la fleur dessinée ainsi que de quelques lignes de guidage fournies par l'utilisateur. Elle utilise ces informations, combinées à une base de données d'éléments botaniques 3D préexistante, afin d'identifier chaque composant de la fleur dessinée par l'utilisateur et d'en proposer une reconstruction 3D. Par ailleurs, les modèles produits ne sont pas botaniquement corrects et suivent une simplification identique à celle que effectuée lors de son esquisse, c'est-à-dire la simplification des détails géométriques, l'arrangement répété des pétales et l'omission de certaines parties telle que le calice. Par conséquent, notre méthode n'est pas utilisable dans le domaine biologique qui nécessite des modèles qui respectent strictement la structure botanique. Cependant, du fait que le temps de reconstruction nécessaire pour produire des modèles de bonne qualité soit relativement court et de la possibilité de créer de nombreuses fleurs d'espèces diverses, notre méthode est tout à fait adaptée à la création de modèles pour la population d'environnement virtuel, tel que des jeux vidéo, où l'exactitude botanique n'est pas nécessaire du moment que les modèles sont réalistes.

Cette méthode comble le défaut principal de la méthode *Floral diagrams and inflorescences* de Ijiri et al. [IOOI05], à savoir le temps nécessaire à la reconstruction. Elle présente également les avantages principaux que doivent avoir les méthodes de reconstruction à l'aide de croquis : seul un dessin relativement abstrait est nécessaire, l'utilisateur n'a aucune manipulation 3D à effectuer et le modèle est produit rapidement, rendant ainsi accessible à n'importe qui la modélisation d'objets floraux de manière quasi instantanée. Cependant, et comme toutes les méthodes se basant sur des a priori, sa grande faiblesse provient de la limitation sur la nature des objets reconstructibles, d'autant plus que notre système repose sur une base de données d'éléments 3D.

La seconde méthode pour répondre au problème de reconstruction géométrique à partir de croquis d'une façon plus générale que notre première approche, c'est-à-dire sans être limitée à une seule catégorie d'objet. Elle a donc été conçue dans le but de proposer une solution permettant de générer n'importe quelle forme 3D à partir d'un croquis d'un seul point de vue. De manière à obtenir les informations nécessaires pour produire des résultats, il est demandé à l'utilisateur d'effectuer son croquis dans le style *descriptif*. De cette façon, la problématique liée aux parties cachées est extrêmement simplifiée, et le programme obtient suffisamment d'informations pour être capable de déduire l'ordonnement en profondeur des éléments. De plus, notre méthode dispose d'un ensemble d'outils simples permettant à l'utilisateur de guider et modifier la reconstruction, ce qui nous assure que le résultat obtenu correspond aux attentes de celui-ci. Ces outils permettent également à l'artiste de corriger le système lorsque celui-ci propose une mauvaise solution ou fait face à des cas ambigus.

L'utilisation d'un style de dessin plus informatif couplé aux outils de guidages fournis à l'utilisateur permet à cette méthode de s'affranchir de toutes formes d'a priori, et donc de produire à partir d'un seul croquis des modèles de tout type d'objets, qu'ils soient organiques ou artificiels. Elle ne fait pas non plus d'hypothèses sur l'angle de vue utilisé par l'artiste, et est donc capable de fonctionner peu importe le point de vue choisi pour l'esquisse, bien que certains soient moins propices à la production de bons résultats. Cette solution présente donc les caractéristiques recherchées dans la reconstruction à base de croquis, c'est-à-dire qu'elle permet de réaliser la reconstruction 3D de tout type d'objets à partir d'un seul croquis et cela d'une façon plus simple et plus rapide qu'avec un logiciel de modélisation 3D classique. Cependant, puisque la reconstruction est guidée par l'utilisateur, notre système n'est pas purement automatique et nécessite des interactions avec l'artiste qui requièrent parfois d'être effectuées dans l'espace 3D, même si celles-ci sont minimales. De ce fait, notre système ne s'affranchit pas totalement des modeleurs 3D classiques puisqu'il partage avec eux des outils 3D communs.

En conclusion, ce manuscrit présente les deux approches que nous avons mises au point au cours de ces 3 années de thèse et qui visent à apporter de nouvelles solutions au problème de reconstruction géométrique à partir de croquis : l'une étant complètement automatique et spécialisée dans la reconstruction de fleurs, l'autre étant interactive, mais permettant la reconstruction de tout type d'objets. Comme on peut le constater, aucune

de nos méthodes ne permet la reconstruction pleinement automatique de formes libres. Le problème de reconstruction géométrique à partir de croquis reste donc ouvert.

## 5.2 Perspective de recherche

Une perspective de recherche pour le futur serait de réussir à automatiser la reconstruction de notre seconde méthode. Une solution envisageable serait d'utiliser les réseaux de neurones pour parvenir à combler les problèmes auxquels notre système actuel doit faire face. À l'heure actuelle, aucune méthode basée sur les CNN n'utilise en entrée un croquis dessiné dans un style plus informatif tel que le style *descriptif*, la plupart des systèmes utilisant de multiples croquis de l'objet à reconstruire comme entrées. Une architecture du réseau en 2 parties serait envisageable. L'idée serait que la première partie analyse le dessin produit par l'utilisateur et propose une décomposition en éléments accompagnée de règles de reconstructions à suivre et que la seconde partie effectue la reconstruction en suivant ces règles. Ainsi la première partie prendrait en entrée un dessin en style descriptif, et chercherait à produire en sortie une décomposition de ce dessin en éléments 2D avec un ordre de profondeur associé à chacun d'entre eux, cette partie serait en réalité assez proche de ce que fait déjà notre méthode, mais l'apprentissage des réseaux de neurones pourrait permettre de résoudre automatiquement les cas ambigus qui demandent une intervention manuelle avec notre méthode actuelle. Pour ce faire, il serait nécessaire de produire un ensemble de données d'apprentissage sous la forme *dessin - ensemble d'éléments composants le modèle final* contenant des cas simples et des cas ambigus, de manière à entraîner le réseau efficacement. La seconde partie du système prendrait en données d'entrée l'ensemble des éléments obtenus par la première partie, ainsi que le dessin initial de l'utilisateur, et serait chargée de produire le résultat 3D final. Ce résultat devra être produit de manière à répondre aux règles de reconstruction produites par le premier sous-réseau tout en suivant les règles de perceptions visuelles que nous utilisons (tel que la compacité).

## Annexe A

### Autres reconstructions de fleurs



FIGURE A.1 – Résultats supplémentaires obtenus avec notre système dédié à la reconstruction de fleurs.

## Annexe B

### Exemples d'apprentissage

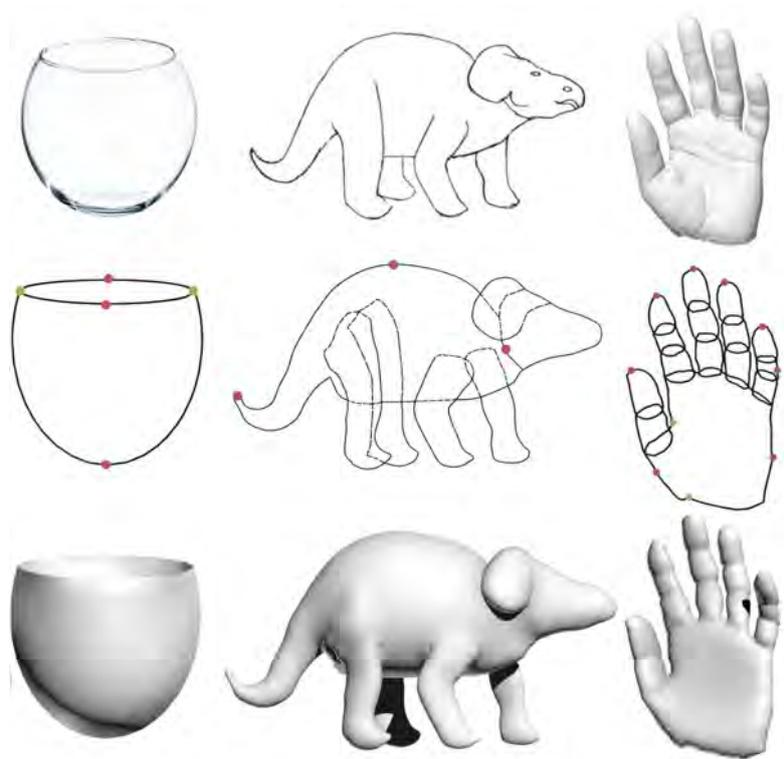


FIGURE B.1 – Les trois exemples d'apprentissage montrés à l'utilisateur durant la session d'entraînement de l'étude utilisateur de notre modèleur *Descriptive*. De haut en bas : le modèle à reconstruire, le dessin fournit en entrée avec les spécifieurs (rouges contraintes de positions, vert points d'angle) et enfin le modèle final reconstruit.

## Annexe C

# Résultats obtenus lors de l'étude utilisateur

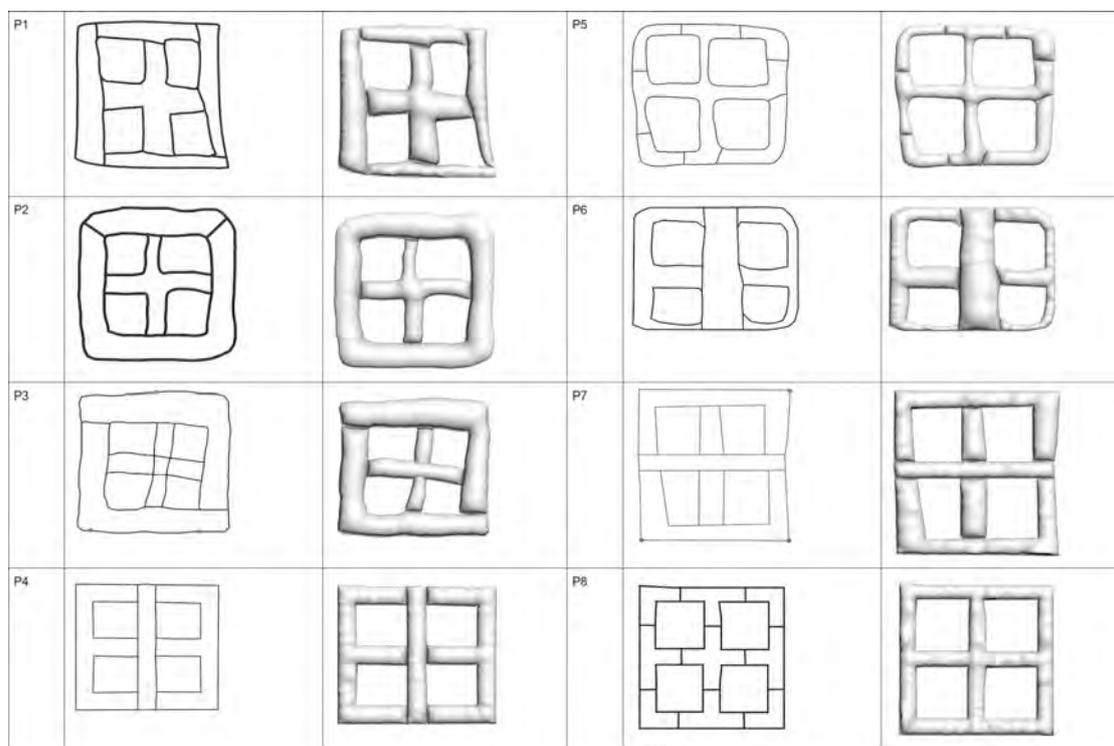


FIGURE C.1 – Résultats obtenus lors du premier essai de nos utilisateurs pour la reconstruction de la fenêtre. Pour chaque participant : son dessin, avec les spécifieurs, ainsi que le modèle reconstruit.

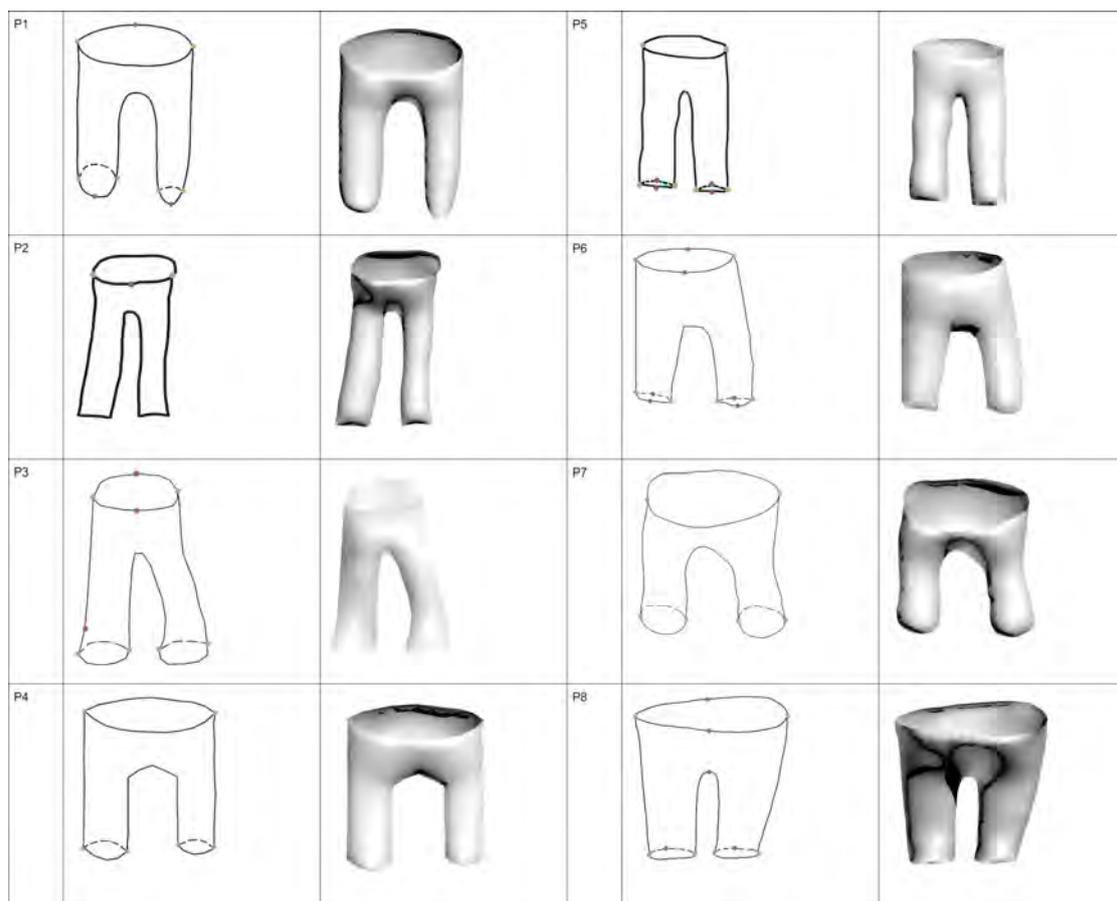


FIGURE C.2 – Résultats obtenus lors du premier essai de nos utilisateurs pour la reconstruction du pantalon. Pour chaque participant : son dessin, avec les spécifieurs, ainsi que le modèle reconstruit.

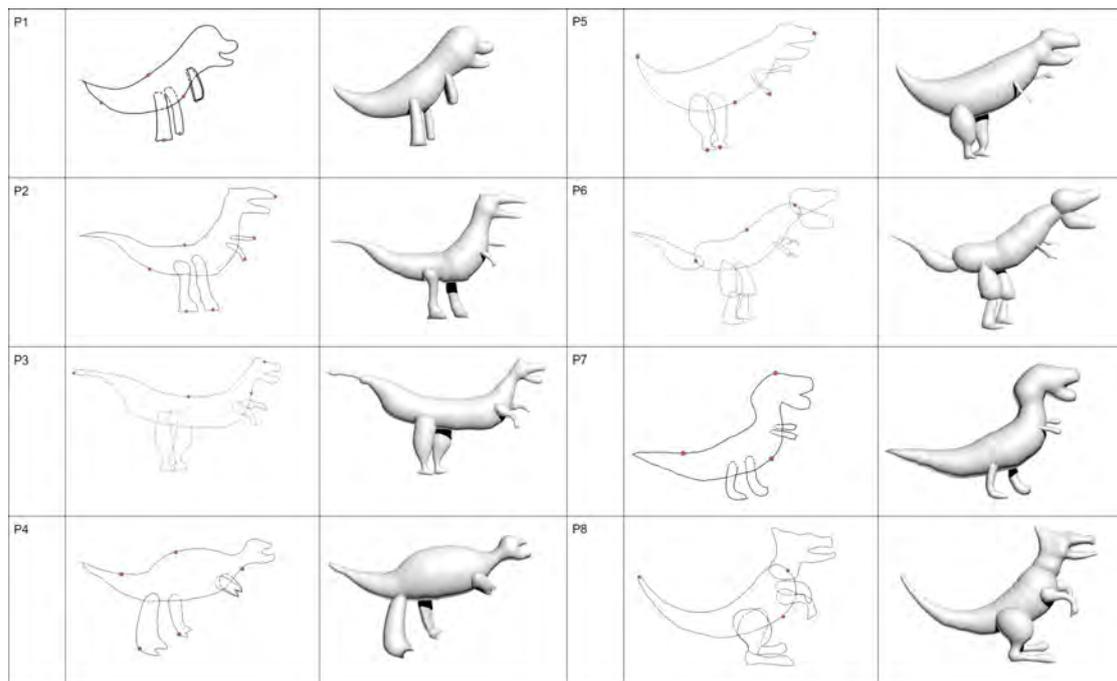
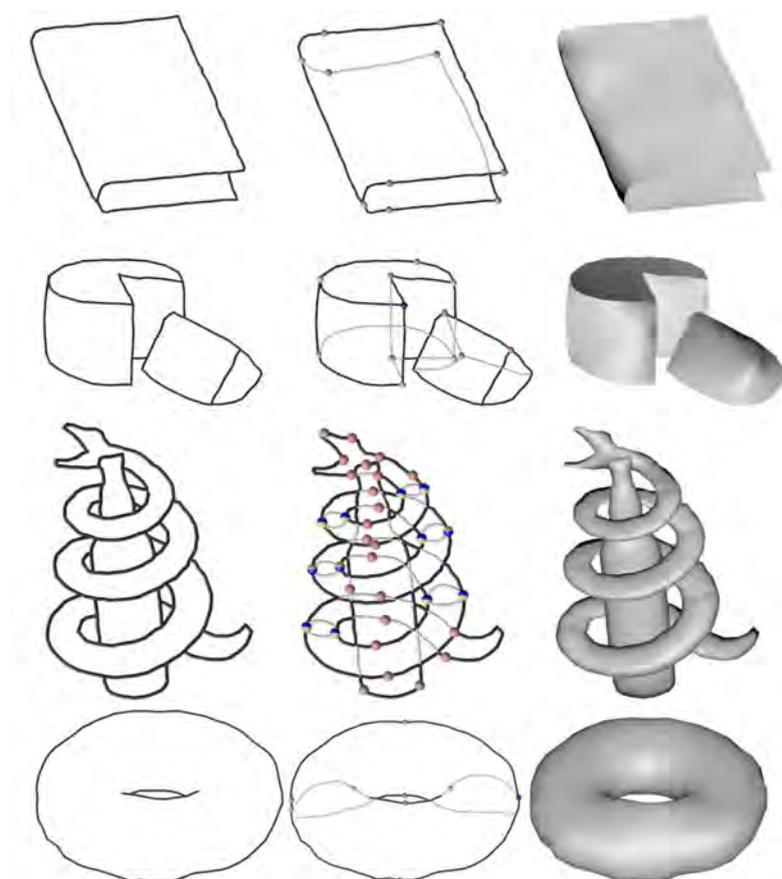


FIGURE C.3 – Résultats obtenus lors du premier essai de nos utilisateurs pour la reconstruction du tyrannosaure. Pour chaque participant : son dessin, avec les spécifieurs, ainsi que le modèle reconstruit.

## Annexe D

# Modèles supplémentaires produits avec Descriptive



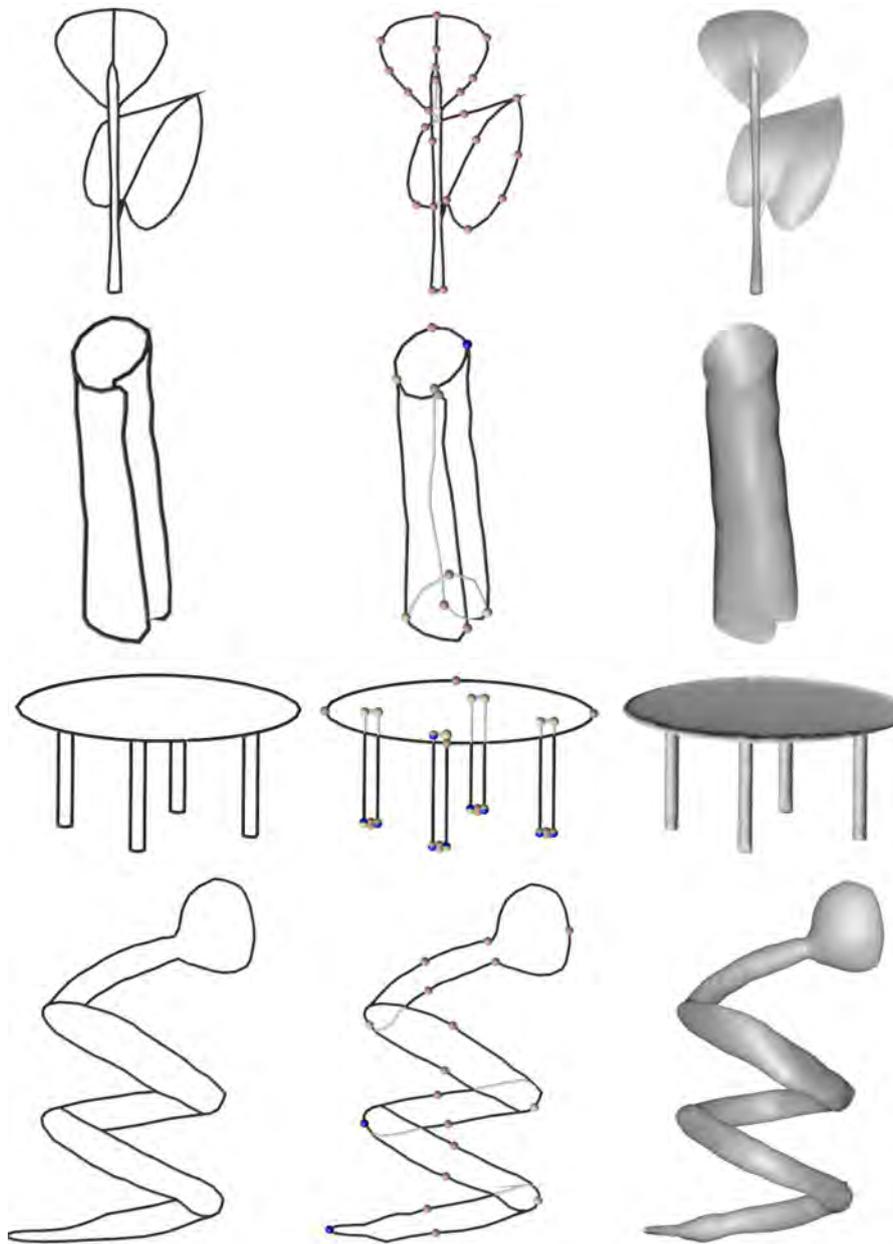


FIGURE D.1 – Modèles produits avec Descriptive. De gauche à droite : le dessin sans parties cachées, le dessin avec les parties cachées et les spécifieurs, le modèle obtenu.

# Bibliographie

- [ACCP03] Brett Allen, Brian Curless, Brian Curless, and Zoran Popović. The space of human body shapes : Reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3) :587–594, July 2003.
- [BBS08] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. Ilovesketch : As-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, UIST '08, pages 151–160, New York, NY, USA, 2008. ACM.
- [BCV<sup>+</sup>15] Mikhail Bessmeltsev, Will Chang, Nicholas Vining, Alla Sheffer, and Karan Singh. Modeling character canvases from cartoon drawings. *Transactions on Graphics (2015)*, 34(5), 2015.
- [BJD<sup>+</sup>12] Peter Borosan, Ming Jin, Doug DeCarlo, Yotam Gingold, and Andrew Nealen. RigMesh : Automatic rigging for part-based shape modeling and deformation. *ACM Transactions on Graphics (TOG)*, 31(6) :198 :1–198 :9, November 2012.
- [BMD13] Philip Buchanan, R. Mukundan, and Michael Doggett. Automatic single-view character model reconstruction. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, SBIM '13, pages 5–14, New York, NY, USA, 2013. ACM.
- [BSCH18] Cédric Bobenrieth, Hyewon Seo, Frédéric Cordier, and Arash Habibi. Reconstructing Flowers from Sketches. *Computer Graphics Forum*, 37(7) :167–178, 2018.
- [Buh03] Martin D. Buhmann. *Radial Basis Functions : Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003.
- [BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [BVS16] Mikhail Bessmeltsev, Nicholas Vining, and Alla Sheffer. Gesture3d : posing 3d characters via gesture drawings. *ACM Trans. Graph.*, 35 :165 :1–165 :13, 2016.

- [CA09] Matthew T. Cook and Arvin Agah. A survey of sketch-based 3-d modeling techniques. *Interact. Comput.*, 21(3) :201–211, July 2009.
- [CiRL<sup>+</sup>16] Byungkuk Choi, Roger Blanco i Ribera, J. P. Lewis, Yeongho Seol, Seokpyo Hong, Haegwang Eom, Sunjin Jung, and Junyong Noh. Sketchimo : Sketch-based motion editing for articulated characters. *ACM Trans. Graph.*, 35(4) :146 :1–146 :12, July 2016.
- [CS07] F. Cordier and H. Seo. Free-form sketching of self-occluding objects. *IEEE Computer Graphics and Applications*, 27(1) :50–59, Jan 2007.
- [CSPyN11] Frederic Cordier, Hyewon Seo, Jinho Park, and Jun yong Noh. Sketching of mirror-symmetric shapes. *IEEE Transactions on Visualization and Computer Graphics*, 17 :1650–1662, 2011.
- [DAI<sup>+</sup>18] Johanna Delanoy, Mathieu Aubry, Phillip Isola, Alexei Efros, and Adrien Bousseau. 3d sketching using multi-view deep volumetric prediction. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 1(21), may 2018.
- [DL16] Chao Ding and Ligang Liu. A survey of sketch based modeling systems. *Frontiers of Computer Science*, 10(6) :985–999, Dec 2016.
- [DNJ<sup>+</sup>18] Marek Dvorožňák, Saman Sepehri Nejad, Ondřej Jamriška, Alec Jacobson, Ladislav Kavan, and Daniel Šýkora. Seamless reconstruction of part-based high-relief models from hand-drawn images. In *Proceedings of International Symposium on Sketch-Based Interfaces and Modeling*, 2018.
- [EBC<sup>+</sup>15] Even Entem, Loïc Barthe, Marie-Paule Cani, Frederic Cordier, and Michiel Van De Panne. Modeling 3D animals from a side-view sketch. *Computers and Graphics*, 46 :221–230, February 2015. Shape Modeling International 2014.
- [EPB<sup>+</sup>19] Even Entem, Amal dev Parakkat, Loïc Barthe, Ramanathan Muthuganapathy, and Marie-Paule Cani. Automatic structuring of organic shapes from a single drawing. *Computers and Graphics*, March 2019.
- [ERB<sup>+</sup>12] Mathias Eitz, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. Sketch-based shape retrieval. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4) :31 :1–31 :10, 2012.
- [FF70] Andrew Fitzgibbon and Robert Fisher. A buyer’s guide to conic fitting. 02 1970.
- [FKS<sup>+</sup>04] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, August 2004.
- [GCR13] Martin Guay, Marie-Paule Cani, and Rémi Ronfard. The line of action : an intuitive interface for expressive character posing. *ACM Trans. Graph.*, 32 :205 :1–205 :8, 2013.

- [GFK<sup>+</sup>18] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet : A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [GIZ09] Yotam Gingold, Takeo Igarashi, and Denis Zorin. Structured annotations for 2D-to-3D modeling. *ACM Transactions on Graphics (TOG)*, 28(5) :148, 2009.
- [Gon95] Jacek Gondzio. Multiple centrality corrections in a primal–dual method for linear programming. *Computational Optimization and Applications*, 6 :137–, 11 1995.
- [HF06] Kai Hormann and Michael S. Floater. Mean value coordinates for arbitrary planar polygons. *ACM Trans. Graph.*, 25(4) :1424–1441, October 2006.
- [HGY17] Xiaoguang Han, Chang Gao, and Yizhou Yu. Deepsketch2face : A deep learning based sketching system for 3d face and caricature modeling. *CoRR*, abs/1706.02042, 2017.
- [HKYM17] H. Huang, E. Kalogerakis, E. Yumer, and R. Mech. Shape synthesis from sketches via procedural models and convolutional networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(8) :2003–2013, Aug 2017.
- [HMC<sup>+</sup>15] Fabian Hahn, Frederik Mutzel, Stelian Coros, Bernhard Thomaszewski, Maurizio Nitti, Markus Gross, and Robert W. Sumner. Sketch abstractions for character posing. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA '15*, pages 185–191, New York, NY, USA, 2015. ACM.
- [HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [IBB15] Emmanuel Iarussi, David Bommes, and Adrien Bousseau. Bendfields : Regularized curvature fields from rough concept sketches. *ACM Transactions on Graphics*, 2015.
- [IIMT07] Takeo Igarashi, Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy : A sketching interface for 3d freeform design. In *ACM SIGGRAPH 2007 Courses, SIGGRAPH '07*, New York, NY, USA, 2007. ACM.
- [IOI06] Takashi Ijiri, Shigeru Owada, and Takeo Igarashi. Seamless integration of initial sketching and subsequent detail editing in flower modeling. *Comput. Graph. Forum*, 25 :617–624, 2006.
- [IOOI05] Takashi Ijiri, Shigeru Owada, Makoto Okabe, and Takeo Igarashi. Floral diagrams and inflorescences : Interactive flower modeling using botanical structural constraints. *ACM Trans. Graph.*, 24(3) :720–726, July 2005.
- [IYYI14] Takashi Ijiri, Shin Yoshizawa, Hideo Yokota, and Takeo Igarashi. Flower modeling via x-ray computed tomography. *ACM Trans. Graph.*, 33(4) :48 :1–48 :10, July 2014.

- [KH06] Olga A. Karpenko and John F. Hughes. Smoothsketch : 3d free-form shapes from complex sketches. *ACM Trans. Graph.*, 25 :589–598, 2006.
- [KHS10] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3D Mesh Segmentation and Labeling. *ACM Transactions on Graphics*, 29(3), 2010.
- [KYC<sup>+</sup>17] Vojtěch Krs, Ersin Yumer, Nathan Carr, Bedrich Benes, and Radomír Měch. Skipppy : Single view 3d curve interactive modeling. *ACM Trans. Graph.*, 36(4) :128 :1–128 :12, July 2017.
- [KYY<sup>+</sup>15] Ismail Khalid Kazmi, Lihua You, Xiaosong Yang, Xiaogang Jin, and Jian J. Zhang. Efficient sketch-based creation of detailed character models through data-driven mesh deformations. *Comput. Animat. Virtual Worlds*, 26(3-4) :469–481, May 2015.
- [LF08] Jeehyung Lee and Thomas Funkhouser. Sketch-based search and composition of 3D models. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*, June 2008.
- [LG13] Zohar Levi and Craig Gotsman. Artisketch : A system for articulated sketch modeling. *Comput. Graph. Forum*, 32 :235–244, 2013.
- [LGK<sup>+</sup>17] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhansu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, 2017.
- [LGY<sup>+</sup>06] Qilei Li, Weidong Geng, Tao Yu, XiaoJie Shen, Newman Lau, and Gwanghyun Yu. Motionmaster : authoring and choreographing kung-fu motions by sketch drawings. In *Symposium on Computer Animation*, 2006.
- [LHZ<sup>+</sup>16] Lei Li, Zhe Huang, Changqing Zou, Chiew-Lan Tai, Rynson W. H. Lau, Hao Zhang, Ping Tan, and Hongbo Fu. Model-driven sketch reconstruction with structure-oriented retrieval. In *SIGGRAPH ASIA 2016 Technical Briefs*, SA '16, pages 28 :1–28 :4, New York, NY, USA, 2016. ACM.
- [Llo82] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Trans. Information Theory*, 28 :129–136, 1982.
- [LLZ<sup>+</sup>17] Yuwei Li, Xi Luo, Youyi Zheng, Pengfei Xu, and Hongbo Fu. Sweepcanvas : Sketch-based 3d prototyping on an rgb-d image. In *UIST*, 2017.
- [LPL<sup>+</sup>17] Changjian Li, Hao Pan, Yang Liu, Alla Sheffer, and Wenping Wang. Bend-sketch : Modeling freeform surfaces through 2d sketching. *ACM Trans. Graph. (SIGGRAPH)*, 36(4) :125 :1–125 :14, 2017.
- [LPL<sup>+</sup>18] Changjian Li, Hao Pan, Yang Liu, Alla Sheffer, and Wenping Wang. Robust flow-guided neural prediction for sketch-based freeform surface modeling. *ACM Trans. Graph. (SIGGRAPH ASIA)*, 37(6) :238 :1–238 :12, 2018.
- [LPS08] Yunfeng Li, Zygmunt Pizlo, and Robert Steinman. A computational model that recovers the 3d shape of an object from a single 2d retinal representation. *Vision research*, 49 :979–91, 08 2008.

- [Mar77] D. Marr. Analysis of occluding contour. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 197(1129) :441–475, 1977.
- [Mar03] Chris Maraffi. *Maya® Character Creation : Modeling and Animation Controls*. New Riders Publishing, Thousand Oaks, CA, USA, first edition, 2003.
- [MHZ<sup>+</sup>15] Yongwei Miao, Feixia Hu, Xudong Zhang, Jiazhou Chen, and Renato Pajarola. Symmsketch : Creating symmetric 3d free-form shapes from 2d sketches. *Computational Visual Media*, 1 :3–16, 2015.
- [NGDA<sup>+</sup>16] Gen Nishida, Ignacio Garcia-Dorado, Daniel G. Aliaga, Bedrich Benes, and Adrien Bousseau. Interactive sketching of urban procedural models. *ACM Trans. Graph.*, 35(4) :130 :1–130 :11, July 2016.
- [NISHA07] Andrew Nealen, Takeo Igarashi, Olga Sorkine-Hornung, and Marc Alexa. Fibermesh : designing freeform surfaces with 3d curves. *ACM Trans. Graph.*, 26 :41, 2007.
- [Noé17] Estelle Noé. *3D layered articulated object from a single 2D drawing*. dissertation, 2017.
- [NW06] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.
- [OBP<sup>+</sup>13] A. Cengiz Öztireli, Ilya Baran, Tiberiu Popa, Boris Dalstein, Robert W. Sumner, and Markus Gross. Differential blending for expressive sketch-based posing. In *Proceedings of the 2013 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, New York, NY, USA, 2013. ACM.
- [OSSJ09] Luke Olsen, Faramarz F. Samavati, Mario Costa Sousa, and Joaquim A. Jorge. Sketch-based modeling : A survey. *Computers and Graphics*, 33 :85–103, 2009.
- [Pal75] Stephen E. Palmer. The effects of contextual scenes on the identification of objects. *Memory & Cognition*, 3(5) :519–526, Sep 1975.
- [PFZVG10] Mukta Prasad, Andrew Fitzgibbon, Andrew Zisserman, and Luc Van Gool. Finding nemo : Deformable object class modelling using curve matching. pages 1720–1727, 06 2010.
- [PJBF00] João Paulo Pereira, Joaquim A. Jorge, Vasco A. Branco, and Fernando Nunes Ferreira. Towards calligraphic interfaces : Sketching 3d scenes with gestures and context icons. In *WSCG*, 2000.
- [PLS<sup>+</sup>15] Hao Pan, Yang Liu, Alla Sheffer, Nicholas Vining, Changjian Li, and Wenping Wang. Flow aligned surfacing of curve networks. *ACM Trans. Graph. (SIGGRAPH)*, 34(4), 2015.
- [PS15] Chris De Paoli and Karan Singh. Secondskin : sketch-based construction of layered 3d models. *ACM Trans. Graph.*, 34(4) :126 :1–126 :10, 2015.

- [RDC10] Louis Ronse De Craene. Floral diagrams : An aid to understanding flower morphology and evolution. *Floral Diagrams : An Aid to Understanding Flower Morphology and Evolution*, pages 1–441, 01 2010.
- [RDI10] Alec Rivers, Frédo Durand, and Takeo Igarashi. 3d modeling with silhouettes. *ACM Trans. Graph.*, 29(4) :109 :1–109 :8, July 2010.
- [SAG<sup>+</sup>13] Alex Shtof, Alexander Agathos, Yotam Gingold, Ariel Shamir, and Daniel Cohen-Or. Geosemantic snapping for sketch-based modeling. *Computer Graphics Forum*, 32(2) :245–253, 2013. Proceedings of Eurographics 2013.
- [SBS19] Dmitriy Smirnov, Mikhail Bessmeltsev, and Justin Solomon. Deep sketch-based modeling of man-made shapes. *CoRR*, abs/1906.12337, 2019.
- [SCOL<sup>+</sup>04] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, SGP '04*, pages 175–184, New York, NY, USA, 2004. ACM.
- [She96] Jonathan Richard Shewchuk. Triangle : Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry : Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.
- [SI07] HyoJong Shin and Takeo Igarashi. Magic canvas : interactive design of a 3-d scene prototype from freehand sketches. In *Graphics Interface*, 2007.
- [SI17] Shogo Seki and Takeo Igarashi. Sketch-based 3d hair posing by contour drawings. In *Symposium on Computer Animation*, 2017.
- [SK08] Christopher Summerfield and Etienne Koechlin. A neural representation of prior information during perceptual inference. *Neuron*, 59 :336–347, 2008.
- [SKMW17] Michael Schwärzler, Lisa-Maria Kellner, Stefan Maierhofer, and Michael Wimmer. Sketch-based guided modeling of 3d buildings from oriented photos. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D '17*, pages 9 :1–9 :8, New York, NY, USA, 2017. ACM.
- [SKSK09] Ryan Schmidt, Azam Khan, Karan Singh, and Gord Kurtenbach. Analytic drawing of 3d scaffolds. In *ACM SIGGRAPH Asia 2009 Papers, SIGGRAPH Asia '09*, pages 149 :1–149 :10, New York, NY, USA, 2009. ACM.
- [SKv<sup>+</sup>14] Daniel Sýkora, Ladislav Kavan, Martin Čadík, Ondřej Jamriška, Alec Jacobson, Brian Whited, Maryann Simmons, and Olga Sorkine-Hornung. Ink-and-Ray : Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transaction on Graphics*, 33(2) :16, 2014.
- [SMT04] Hyewon Seo and Nadia Magnenat-Thalmann. An example-based approach to human body manipulation. *Graph. Models*, 66(1) :1–23, January 2004.

- [SN06] Clayton D. Scott and Robert D. Nowak. Robust contour matching via the order-preserving assignment problem. *IEEE Transactions on Image Processing*, 15 :1831–1838, 2006.
- [SWSJ07] R. Schmidt, B. Wyvill, M. C. Sousa, and J. A. Jorge. Shapeshop : Sketch-based solid modeling with blobtrees. In *ACM SIGGRAPH 2007 Courses, SIGGRAPH '07*, New York, NY, USA, 2007. ACM.
- [SXY<sup>+</sup>11] Tianjia Shao, Weiwei Xu, Kangkang Yin, Jingdong Wang, Kun Zhou, and Baining Guo. Discriminative sketch-based 3d model retrieval via robust shape matching. *Comput. Graph. Forum*, 30 :2011–2020, 09 2011.
- [SZ03] J. Sivic and A. Zisserman. Video google : A text retrieval approach to object matching in videos. volume 2, pages 1470 – 1477 vol.2, 11 2003.
- [WMW<sup>+</sup>08] B. Watson, P. Muller, P. Wonka, C. Sexton, O. Veryovka, and A. Fuller. Procedural Urban Modeling in Practice. *IEEE Computer Graphics and Applications*, 28(3) :18–26, May 2008.
- [WZS19] Kai Wang, Jianmin Zheng, and Hock Soon Seah. Progressive sketching with instant previewing. *Computers and Graphics*, 81 :9 – 19, 2019.
- [XCF<sup>+</sup>13] Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. Sketch2scene : Sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics*, 32(4) :123 :1–123 :12, 2013.
- [XCS<sup>+</sup>14] Baoxuan Xu, William Chang, Alla Sheffer, Adrien Bousseau, James McCrae, and Karan Singh. True2form : 3d curve networks from 2d sketches via selective regularization. *Transactions on Graphics (Proc. SIGGRAPH 2014)*, 33(4), 2014.
- [XFZ<sup>+</sup>19] P. Xu, H. Fu, Y. Zheng, K. Singh, H. Huang, and C. Tai. Model-guided 3d sketching. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2019.
- [XXM<sup>+</sup>13] Xiaohua Xie, Kai Xu, Niloy J. Mitra, Daniel Cohen-Or, Wenyong Gong, Qi Su, and Baoquan Chen. Sketch-to-design : Context-based part assembly. *Computer Graphics Forum*, xx(xx) :xx, 2013.
- [YGCO<sup>+</sup>14] Feilong Yan, Minglun Gong, Daniel Cohen-Or, Oliver Deussen, and Baoquan Chen. Flower reconstruction from a single photo. *Comput. Graph. Forum*, 33 :439–447, 2014.
- [YHJ<sup>+</sup>17] C. Yeh, S. Huang, P. K. Jayaraman, C. Fu, and T. Lee. Interactive high-relief reconstruction for organic and double-sided objects from a photo. *IEEE Transactions on Visualization and Computer Graphics*, 23(7) :1796–1808, July 2017.
- [ZHH96] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch : an interface for sketching 3d scenes. In *SIGGRAPH Courses*, 1996.
- [ZYFY14] C. Zhang, M. Ye, B. Fu, and R. Yang. Data-driven flower petal modeling with botany priors. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 636–643, June 2014.

- [ZZCJ13] Yixin Zhuang, Ming Zou, Nathan Carr, and Tao Ju. A general and efficient method for finding cycles in 3d curve networks. *Acm Transactions on Graphics, (Siggraph Asia 2013)*, 32(6) :1–10, 2013.