



## Reconstruction de modèles virtuels de vignes à partir d'images

Jérôme Guénard, Géraldine Morin, Frédéric Boudon, Pierre Gurdjos, Vincent Charvillat

► **To cite this version:**

Jérôme Guénard, Géraldine Morin, Frédéric Boudon, Pierre Gurdjos, Vincent Charvillat. Reconstruction de modèles virtuels de vignes à partir d'images. ORASIS - Congrès des jeunes chercheurs en vision par ordinateur, Jun 2011, Praz-sur-Arly, France. 2011. <inria-00595483>

**HAL Id: inria-00595483**

**<https://hal.inria.fr/inria-00595483>**

Submitted on 24 May 2011

**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.

# Reconstruction de modèles virtuels de vignes à partir d'images

Jérôme Guénard<sup>1</sup> Géraldine Morin<sup>1</sup> Frédéric Boudon<sup>2</sup> Pierre Gurdjos<sup>1</sup> Vincent Charvillat<sup>1</sup>

<sup>1</sup> IRIT - VORTEX - Université de Toulouse

<sup>2</sup> INRIA - Cirad - Montpellier

{jguenard, morin, pgurdjos, charvi}@enseeiht.fr ; frederic.boudon@cirad.fr

## Résumé

*Nous présentons ici une méthode de reconstruction de modèles virtuels de pieds de vignes à partir d'images. Pour cela, nous utilisons une technique d'analyse par synthèse qui consiste à caractériser une image à partir d'un certain nombre de connaissances a priori sur la scène 3D. Tout d'abord, à partir d'une première analyse des images, nous obtenons une approximation initiale de la plante modélisée en 3D. Puis, nous affinons cette modélisation par un processus d'optimisation où la projection du modèle reconstruit est comparée avec l'image d'origine. Pour être efficace, notre méthode n'optimise pas le positionnement de chacune des feuilles mais un feuillage réaliste et cohérent avec les images.*

## Mots Clef

Modélisation de plantes, analyse par synthèse, traitement d'images, reconstruction 3D.

## Abstract

*We propose a method for reconstructing virtual model of vines from images. For this, an analysis by synthesis method is used and consist in characterizing an image using a number of a priori knowledge about the 3D scene. Initially, we get an approximation of the plant modeled in 3D. Then, comparing its reprojection with the original image, we refine this model through an iterative optimisation process. To be efficient, our method do not optimize positioning of individual leaves but rather a realistic foliage consistent with the images.*

## Keywords

Plant modelisation, analysis-by-synthesis, image processing, 3D reconstruction.

## 1 Introduction

La modélisation de plantes est une tâche importante aussi bien en infographie, pour créer des environnements virtuels réalistes [DL05][BMG06], qu'en biologie pour comprendre le fonctionnement des plantes [GS05]. Ce travail a pour objectif de créer un modèle de plant de vigne à partir de plusieurs photographies [GGM\*10]. Il s'inscrit dans

le cadre d'un projet (Section 6, page 7) dont l'objet est la mesure des caractéristiques globales et locales de la vigne et leur corrélation avec le rendement et/ou la qualité du vin produit.

Nous supposons que les propriétés géométriques du plant de vigne sont contraintes par ses propriétés biologiques intrinsèques et les techniques agricoles liées à leur culture, telles que la taille ou le palissage. L'approche que nous utilisons pour la modélisation d'un plant virtuel est l'*analyse par synthèse* (Figure 1, page 2). Celle-ci a pour but d'expliquer un ensemble de données (ici des photographies) le mieux possible avec le plus petit ensemble possible de *paramètres* qui le caractérisent. Dans la suite, on distinguera deux modèles différents. Le premier, que l'on appellera *modèle génératif*, détermine la manière dont les paramètres génèrent les photographies. Le deuxième est le *modèle descriptif*. C'est le modèle inverse qui, à partir des photographies, retrouve les paramètres qui l'ont caractérisé. L'idée est alors d'effectuer une reconstruction du plant à partir des photographies, tout en s'aidant du modèle génératif.

**État de l'art.** De nombreux travaux portent sur la génération de maquettes réalistes de plantes. Beaucoup sont basés sur le formalisme de simulation des *L-systèmes* [PL90]. Un L-système est une grammaire formelle qui permet de modéliser le développement d'une plante. L'idée principale des L-systèmes consiste à réécrire une chaîne de modules représentant la structure de la plante. Les règles de réécriture expriment la création et le changement d'état des différents modules de la plante à travers le temps. Récemment, Palubicki *et al.* [PHL\*09] simulent avec des L-systèmes le développement et la croissance d'une plante en fonction de son environnement et de la compétition entre ses différents organes pour peupler un volume. Cependant, un modèle ainsi généré est biologiquement réaliste mais ne correspond pas à une plante réellement observée.

Différents travaux s'intéressent à la modélisation des plantes à partir d'images. Certains n'utilisent qu'une seule image [TFX\*08], [ZZZ06]. Dans ces approches, un contour de la plante dans l'image est déterminé de manière interactive afin d'en extraire un squelette. Une représentation 3D est alors déduite puis le feuillage ajouté. Les travaux

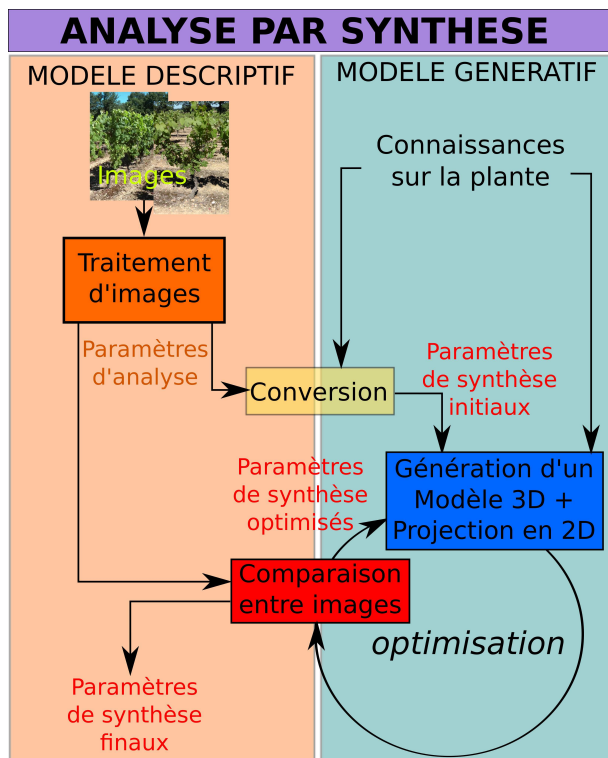


FIGURE 1 – **Notre approche.** À gauche le modèle descriptif qui décrit comment passer de l'image à la scène 3D. À droite, le modèle génératif qui explique comment passer des connaissances sur la scène à une modélisation 3D de celle-ci puis à une projection.

récents de Quan *et al.* [QTZ\*06],[QWTY07] utilisent plusieurs images pour reconstruire un modèle 3D d'arbres ou de plantes. Ils commencent par obtenir un nuage de points quasi-dense à partir de l'analyse des images qu'ils segmentent, afin d'identifier les feuilles. Puis, un modèle paramétrique de feuille est ajusté sur chaque ensemble de points. Ils génèrent ensuite une arborescence à partir d'informations données par l'utilisateur [QTZ\*06] ou à partir d'une procédure générative qui reproduit les mêmes motifs arborescents déduits de la base de la structure jusqu'aux feuilles [QWTY07]. Dans [RMMD04], une autre reconstruction à partir de plusieurs images est proposée, basée sur la notion de *billboards*. Ces travaux utilisent la redondance et les propriétés d'auto-similarité d'une plante pour reconstruire les parties occultées en les supposant ressemblantes aux parties visibles. Dans [NFD07], un volume englobant la plante est d'abord construit sous forme de voxels après une analyse des images. Ensuite des particules remplissent ce volume et leurs chemins dans le volume donnent la géométrie des branches. Dans ce cas, l'utilisateur contribue en spécifiant des courbes donnant la forme générale de la structure arborescente. La plupart de ces travaux nécessitent une intervention humaine lors du traitement, que ce soit pour la segmentation, la création de modèles notamment de feuilles ou encore l'amélioration du rendu final.

Nous nous sommes également appuyés sur certains travaux

d'analyse par synthèse [YK06] [NSH08] [TCYZ03]. Le but de cette approche est de « comprendre » une image en la décomposant à différents niveaux. Dans [YK06], Yuille *et al.* font l'hypothèse qu'un processus de vision *bas-niveau* (détection de primitives, segmentation couleur, etc.) est ambigu alors qu'un processus de vision *haut-niveau* (détection d'objets par apprentissage, utilisation du contexte, etc.) ne l'est pas. Ces auteurs travaillent donc d'abord sur l'image avec des processus bas-niveau. Le modèle fait alors des premières propositions sur ce qu'il comprend de l'image (comme la détection d'un objet ou la structure de la scène) avec des probabilités qui mesurent un degré de certitude. Si la probabilité est suffisamment élevée, la proposition est acceptée, sinon, elle est soumise à une nouvelle analyse de l'image avec un processus plus haut niveau. Cela permet d'avoir un algorithme qui ne soit pas trop long en temps de calcul car les processus haut-niveau ne sont pas systématiquement utilisés. On peut aussi représenter une image par un graphe de scène où différents niveaux détaillent des parties plus ou moins fines de l'image [TCYZ03]. Le premier niveau du graphe correspond, par exemple, aux personnes en premier plan ou à certains objets prédominants. Le niveau suivant décompose ces premières détections en différents textures, texte, sous-objets. On continue tant que la résolution de l'image le permet. Dans ces travaux, il est ensuite possible de re-synthétiser l'image à partir de l'interprétation que l'on en a faite. Il est alors envisageable d'affiner cette interprétation en comparant l'image générée avec l'image d'origine. Par exemple, dans [NSH08], Nair *et al.* ont créé un modèle paramétré d'œil humain avec seulement 8 paramètres par une étape d'apprentissage à partir d'images réelles. Ensuite, à partir d'une nouvelle image et via un processus itératif, ils cherchent à retrouver les meilleurs paramètres possibles pour modéliser l'œil photographié. Cette idée générale inspire fortement le travail présenté dans cet article.

**Description de notre méthode.** Nous utilisons une méthode d'analyse par synthèse utilisant deux modèles que nous décrivons brièvement (Figure 1, page 2).

Le premier modèle, appelé modèle génératif, prend en entrée des connaissances sur la plante considérée et retourne en sortie un modèle virtuel 3D. Dans notre cas, il s'agit de pied de vigne. Les techniques de palissage utilisées contraignent fortement l'architecture générale des pieds. Ces connaissances *a priori* permettent de simplifier la génération du modèle 3D. Ce qui différencie un pied d'un autre est contrôlé par des paramètres dit *paramètres de synthèse*. Ce modèle a été défini grâce au modelleur L-Py [BCPG10] basé sur les *L-systèmes* et le langage Python. L'utilisation des *L-systèmes* est particulièrement justifiée ici car ceux-ci permettent de générer des plantes de manière récursive en utilisant des règles de base simples. Des outils intégrés au modelleur permettent ensuite de projeter le modèle résultant pour former une image.

Le deuxième modèle, appelé modèle descriptif, a pour but

de modéliser une plante existante à partir d'images de celle-ci. Pour cela, nous procédons en deux étapes. Tout d'abord, une première étape consiste à effectuer un traitement sur les images. Des *paramètres d'analyse* sont extraits. Ces paramètres peuvent, par exemple, être la région d'une image représentant le feuillage de la plante (Figure 4, page 4). Ils sont ensuite convertis en paramètres de synthèse en utilisant les connaissances sur la plante. Par exemple, un squelette est calculé à partir de la segmentation du feuillage (Figure 9, page 6). Ces paramètres de synthèse sont ensuite fournis au modèle génératif, précédemment construit, afin d'obtenir une première modélisation de la plante puis, une image est calculée par projection. La deuxième étape sert donc à affiner les paramètres de synthèse trouvés lors de la première étape. Pour cela, une étape d'optimisation vient minimiser la différence entre l'image d'origine et l'image reprojétée. Une métrique quantifiant les différences est donc explicitée.

### Plan de l'article

Dans la partie 2, nous présentons le modèle génératif et les paramètres de synthèse dont il dépend. Puis, dans la partie 3, nous détaillons la partie traitement des images qui consiste à extraire les paramètres d'analyse pour les transformer en paramètres de synthèse et ainsi, obtenir une première estimation de la modélisation du pied de vigne que l'on souhaite reconstruire. La partie 3.2 commente la minimisation que nous utilisons pour parvenir à une modélisation de la plante la plus satisfaisante possible.

## 2 Modèle génératif

Le modèle génératif permet de synthétiser une modélisation 3D de la scène photographiée. Dans notre cas, nous nous intéressons aux scènes représentant des pieds de vigne. Dans la nature, les plantes sont des éléments géométriques organisés et il est donc possible de modéliser une plante avec un certain nombre de règles. Tout degré de liberté pouvant expliquer une variation entre deux plantes de même type pourra être associé à un paramètre de synthèse. Ainsi, en faisant seulement varier les paramètres de synthèse, il sera possible de modéliser une variété de plantes d'un même type.

Pour générer notre modèle, nous utilisons les L-systèmes [PL90], particulièrement bien adaptés dans notre cas, via le modèleur L-py [BCPG10]. En effet, le modèleur L-Py utilise une combinaison entre des constructions L-systèmes et le langage Python pour modéliser des objets naturels.

Nous avons choisi de générer notre modèle en deux étapes i.e. la structure ramifiée puis le feuillage. Nous supposons que la structure ramifiée d'un pied de vigne est contenue dans un plan. Cette hypothèse est justifiée car le branchage est attaché à des fils de fer, placés dans le plan médian du rang par le cultivateur. De plus, elle est toujours composée d'un tronc d'où part horizontalement une branche principale appelée la baguette. En effet, cette branche est guidée par le vigneron le long d'un fil de fer. Enfin, des

rameaux poussent à partir de cette baguette. En général, ils sont au nombre de deux ou trois. La structure topologique de ce modèle est donc très simple. Cependant, dans le but d'adapter notre méthode à d'autres types de plantes ou à des cas particuliers de vignes, nous avons choisi un modèle un peu plus générique. Ainsi, chaque branche sera paramétrée par un certain nombre de nœuds 2D qui sont les nœuds de contrôle de la branche. A partir de chacun de ces nœuds peut partir une ou plusieurs branches latérales qui sont paramétrées de la même manière que leur branche mère. Nous construisons ainsi une structure arborescente de manière récursive. Ensuite, pour modéliser en 3D cette structure, chaque branche est un cylindre généralisé le long d'une courbe passant par tous ses nœuds. Une courbe B-Spline est construite avec un schéma d'interpolation local de degré 3 [PT97]. Un rayon est également attribué à chacun de ces nœuds pour déterminer le rayon du cylindre généralisé en ces différents nœuds. Ce rayon est interpolé linéairement entre deux nœuds. Des textures extraites d'images réelles sont ensuite appliquées sur les branches (Figure 2.a, page 3). L'intérêt de cette construction est qu'elle nous permet de diminuer le nombre de paramètres pour une branche tout en gardant suffisamment de généralité pour modéliser des structures plus complexes. Par exemple, on sait que le tronc d'un pied de vigne pousse de manière verticale. Il peut alors être paramétré plus simplement par une longueur et éventuellement un nombre d'oscillations. Ainsi, pour l'utiliser dans notre modèle génératif, nous extrayons de cette représentation simple les nœuds 2D permettant de définir cette branche dans le modèleur L-Py. Cette étape est importante si l'on souhaite diminuer le nombre de paramètres lors de la dernière étape d'optimisation.

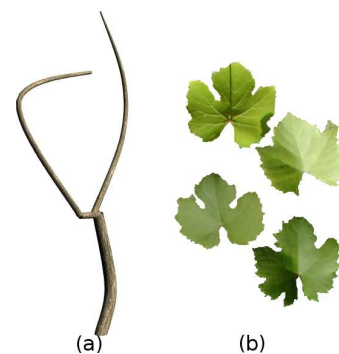


FIGURE 2 – a) Modélisation du branchage d'une plante. b) Textures de feuilles de vigne.

En ce qui concerne le feuillage, nous avons commencé par extraire des textures de feuilles à partir d'images réelles (Figure 2.b, page 3). A chacun des nœuds 2D définissant le branchage est également attribué une valeur correspondant au rayon du cylindre englobant les feuilles en ce nœud. Ainsi, pour modéliser le feuillage, des petites branches de longueur proportionnelle à cette valeur sont générées à partir des branches principales. Sur chacune de ces branchettes

sont alors positionnées des feuilles (entre 2 et 4) modélisées par une surface de Bézier sur laquelle une texture choisie de manière aléatoire est plaquée. Nous obtenons ainsi une modélisation complète du pied de vigne à partir de peu de paramètres de synthèse (Figure 3, page 4).



FIGURE 3 – Exemple de modélisation d'un pied de vigne caractérisé par 28 paramètres de synthèse.

Il suffit alors de reprojeter le modèle 3D généré pour obtenir une image.

### 3 Modèle descriptif

Le modèle descriptif permet de caractériser la plante étudiée à partir d'images de celle-ci. La constitution d'un tel modèle nécessite trois étapes : le traitement des images en vue de l'extraction de paramètres d'analyse, la conversion de ces derniers en paramètres de synthèse et enfin l'étape d'optimisation pour améliorer la qualité de la modélisation.

#### 3.1 Calcul des paramètres d'analyse

Dans un premier temps, il s'agit de réaliser, d'une part, l'extraction du contour du feuillage de la plante et, d'autre part, la détection des branches visibles (souvent que le tronc). Nous allons détailler les différentes techniques de traitement d'images utilisées dans le cas des pieds de vigne.

Lors de nos expérimentations, nous nous sommes déplacés à l'intérieur des rangs de vigne avec un caméscope HD (1920×1080 pixels) et avons fait l'acquisition de séquences d'images. Le mouvement, en caméra portée, a été le plus rectiligne possible, parallèle aux plans du sol et des rangs, en contraignant l'axe optique d'être lui aussi parallèle au plan du sol. La hauteur et l'orientation de la caméra peuvent donc être supposées constantes ; de plus, les paramètres intrinsèques de la caméra ont été pré-calculés, grâce à la procédure de calibrage décrite dans [GsK05]. L'ensemble de ces connaissances *a priori* forme les hypothèses de base dans la suite de notre traitement.

Sous l'hypothèse de la planéité de la structure ramifiée de la plante, dont le plan de support est noté  $\mathcal{P}$ , nous souhaitons réaliser la segmentation du feuillage dans une vue fronto-parallèle à  $\mathcal{P}$ . Nous décrivons dans le paragraphe suivant, comment segmenter le feuillage lorsque les

acquisitions ont été faites, conformément à nos hypothèses, avec un mouvement de translation parallèle à  $\mathcal{P}$ . Dans le cas où l'axe optique de la caméra n'est pas perpendiculaire à  $\mathcal{P}$ , nous expliquons comment se ramener au cas fronto-parallèle.

#### Segmentation du feuillage

Pour effectuer l'extraction du feuillage du premier rang, nous avons combiné deux méthodes. Tout d'abord, une approche bayésienne basée sur les composantes RVB nous a permis de séparer le feuillage du ciel et du sol (Figure 4, page 4) grâce à un apprentissage préalable.



FIGURE 4 – Segmentation du feuillage par approche bayésienne sur les composantes RVB. La partie segmentée est représentée en violet sur l'image de droite.

Cependant, cela n'est pas suffisant car le feuillage des rangs à l'arrière plan est confondu avec le feuillage du premier rang (c.-à-d. du rang considéré). Nous utilisons alors un suivi de points d'intérêt pendant la séquence d'images, en utilisant l'algorithme *KLT* [TK91]. Ainsi, une disparité, intrinsèquement corrélée à la profondeur, peut être associée à chaque point suivi. Le mouvement de la caméra étant une translation parallèle au rang de vigne, pour une colonne fixée de l'image, les pixels correspondant à des points du feuillage du premier rang se déplacent plus rapidement que ceux correspondant à des points du feuillage de derrière. La disparité/profondeur d'un point dépend directement de sa vitesse de déplacement et de son indice de colonne dans l'image. En considérant chaque point d'intérêt dans le repère (numéro de colonne, déplacement entre 2 images), nous avons réussi à classifier automatiquement les points de premier rang, à partir d'une approche de type ACP. Ensuite, chaque pixel de l'image est assigné à la même classe que son point d'intérêt le plus proche.

Nous obtenons alors une segmentation satisfaisante (Figure 5, page 5).

#### Rectification métrique dans le cas non fronto-parallèle

Lors de nos expérimentations, à cause de la limitation du champ de vision, les images ne sont pas acquises avec la caméra fronto-parallèle au plan  $\mathcal{P}$  supportant les pieds de vigne d'un même rang. En effet, la profondeur disponible





FIGURE 5 – **Segmentation du feuillage du premier rang** combinant l’approche bayésienne et le suivi des points d’intérêt. Sur l’image de gauche, les points d’intérêt en cyan correspondent à des points du feuillage suffisamment proches pour appartenir à la rangée de devant. L’image de droite montre le résultat obtenu.

dans les rangées de vigne est insuffisante pour avoir assez de recul. Une rectification métrique de l’image doit donc être effectuée pour segmenter le feuillage. Elle consiste à transformer l’image de manière à retrouver les angles et les rapports de longueur du plan  $\mathcal{P}$  et donc le parallélisme des images des droites de  $\mathcal{P}$  parallèles au sol. Puisque la caméra est calibrée, une condition nécessaire et suffisante est de déterminer dans l’image la ligne de fuite  $l_{fb}$  de  $\mathcal{P}$  qui est le lieu de tous les points de fuite de  $\mathcal{P}$ , contenant en particulier le point de fuite  $p_{f1}$  associée à la direction  $\vec{v}$  dans  $\mathcal{P}$  parallèle au sol (Figure 6, page 5).

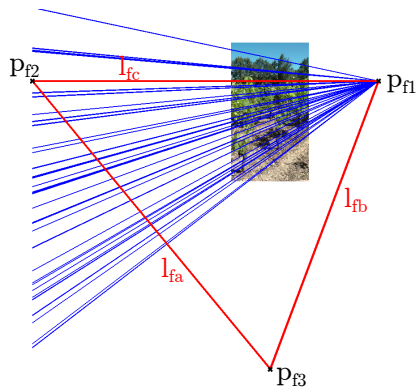


FIGURE 6 – **Trois lignes de fuite** : du plan  $\mathcal{P}$  ( $l_{fb}$ ), du plan du sol ( $l_{fc}$ ) et du plan perpendiculaire à  $\mathcal{P}$  et au sol ( $l_{fa}$ ).

Nous décrivons maintenant comment déterminer le point de fuite  $p_{f1}$  associé à la direction  $\vec{v}$  dans  $\mathcal{P}$  parallèle au sol et comment en déduire la ligne de fuite  $l_{fb}$  de  $\mathcal{P}$ . Pour cela, on utilise une fois encore le suivi des points d’intérêt. Sous les hypothèses de translation de la caméra dans la direction  $\vec{v}$ , on note que le point de fuite  $p_{f1}$  associé à  $\vec{v}$  coïncide, par définition, à l’épipole dans l’image. Ainsi  $p_{f1}$  peut être facilement estimé en estimant la matrice fondamentale à partir de deux images-clés (Figure 7, page 5). Puisque nous avons supposé de plus que l’axe optique de la caméra est

parallèle au sol, on en déduit que la ligne de fuite  $l_{fc}$  du plan du sol est une droite horizontale dans l’image. Ainsi, sachant que  $p_{f1}$  est sur  $l_{fc}$ , on en déduit  $l_{fc}$  très facilement. La matrice de calibrage  $K$  ayant été calculée au préalable, on retrouve le point de fuite  $p_{f3}$  associé à la direction perpendiculaire au sol [HZ00] :

$$p_{f3} = \omega^* l_{fc}$$

où  $\omega^* = KK^T$ . Maintenant que l’on connaît deux points de fuite  $p_{f1}$  et  $p_{f3}$  se situant sur la ligne de fuite associée aux directions de la scène parallèles au plan des branchages, on calcule facilement une homographie de rectification métrique (dont la matrice a pour deux premières colonnes les vecteurs  $p_{f1}$  et  $p_{f3}$  multipliés par des facteurs d’échelle faciles à déterminer connaissant  $\omega^*$ ), puis on l’applique à l’image de façon à « obtenir » une vue fronto-parallèle du feuillage (Figure 8, page 5).



FIGURE 7 – **Suivi des points d’intérêt** (représentés en vert). Les droites portant ces segments verts concourent en un point qui est le point de fuite associé à la direction de la caméra.



FIGURE 8 – **Rectification métrique**. À gauche, l’image originale et à droite, l’image rectifiée.

Nous connaissons maintenant les paramètres d’analyse de la plante à modéliser : d’une part la segmentation du feuillage en fronto-parallèle et d’autre part le contour du tronc extrait grâce à une segmentation couleur par approche

bayésienne, aidée par une connaissance *a priori* sur l'espace entre troncs dans la scène 3D.

### 3.2 Calcul des paramètres de synthèse

Cette partie explique comment retrouver les paramètres de synthèse à fournir au modèle génératif.

#### Initialisation

L'étape d'initialisation est déterminante pour obtenir une modélisation finale satisfaisante de notre plante. Le but est de convertir les paramètres d'analyse extraits lors de la partie précédente en paramètres de synthèse initiaux, tout en utilisant des connaissances sur la plante à modéliser. Étant donné le modèle paramétré de notre plante (Section 2, page 3), la première étape consiste à retrouver la position en 2D des principaux nœuds du branchage de notre plante. Nous rappelons ici que les branches sont supposées être toutes dans un même plan. Ainsi, la largeur de la plante étant suffisamment faible par rapport à notre scène, nous considérons que le contour du feuillage donne une bonne approximation du branchage de départ.

Nous nous sommes inspirés des travaux de Tan *et al.* [TFX\*08] qui utilisent des *points attracteurs* positionnés dans l'image segmentée pour retrouver le branchage d'un arbre. Les méthodes classiques de squelettisation, comme la recherche de l'axe médian, ne sont pas adaptées ici car elles ne prennent pas en compte notre connaissance sur la structure de la plante. De plus, elles ne sont pas robustes aux petites modifications du feuillage ou aux trous.

Nous recherchons un squelette du feuillage, défini par un ensemble de segments de droites formant une ligne polygonale et paramétré par  $h$  sommets  $p_k$ ,  $k = 1..h$ . Nous commençons par placer régulièrement dans le feuillage un ensemble noté  $A = \{a_i, i = 1..n\}$  de points dits attracteurs (Figure 9, page 6). Soit  $S = \{s_j, j = 1..m\}$  un ensemble de points du squelette séparés par une certaine distance prédéfinie et construit à partir de  $\{p_k\}$ .

Le problème de squelettisation est formulé comme le problème d'optimisation suivant

$$\min_{p_k} \mathcal{C}(A, S)$$

avec

$$\mathcal{C}(A, S) = \frac{1}{n} \sum_{a_i \in A} \min_{s_j \in S} \delta^2(a_i, s_j) + \frac{1}{m} \sum_{s_j \in S} \min_{a_i \in A} \delta^2(a_i, s_j)$$

où  $\delta^2(a_i, s_j)$  désigne le carré de la distance euclidienne entre les deux points  $a_i$  et  $s_j$ . On cherche ainsi à trouver un compromis entre l'ajustement des attracteurs à une ligne polygonale formant le squelette (dont les points sont ceux de  $S$ ) et la contrainte que ces points du squelette soient « dans » le feuillage.

Une solution est cherchée au sens des moindres carrés via la méthode de Levenberg-Marquardt (fonction *lsqnonlin* de Matlab), où l'ensemble  $S$  est déduit à chaque itération

des sommets  $p_k$ . Le critère  $\mathcal{C}(A, S)$  est continu mais n'est pas dérivable à cause de la fonction  $\min(\cdot)$ . Pour rendre ce critère dérivable en tout point, on approche  $\min(\cdot)$  par  $\text{softmin}(\cdot)$  en définissant [MBP\*08]

$$\text{softmin}_{x_i \in X} \delta^2(x_i, y) = -\log \sum_{x_i \in X} \exp(-\delta^2(x_i, y))$$

où  $X$  est un ensemble de points et  $y$  un point donné.

Afin d'améliorer notre méthode, nous utilisons deux types d'attracteurs. Les attracteurs du premier type sont placés sur l'image du tronc et n'attirent que les points du squelette appartenant au tronc. Ceux du second sont répartis dans le feuillage et attirent tous les autres points du squelette se trouvant sur le branchage portant les feuilles.

La résolution itérative du problème requiert une solution de départ. Nos expérimentations ont mis en évidence la possibilité de fournir une solution de départ arbitraire (respectant seulement les contraintes) à partir de laquelle l'algorithme converge (Figure 9, page 6).

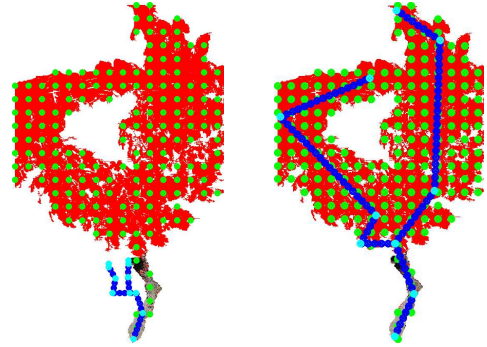


FIGURE 9 – Calcul du squelette du branchage. Les attracteurs  $a_i$ ,  $i = 1..n$  sont représentés en vert, les sommets du squelette  $p_k$ ,  $k = 1..h$  en cyan et les points sur le squelette  $s_j$ ,  $j = 1..m$  en bleu. À gauche, le squelette contraint du départ. À droite, le squelette après minimisation.

On obtient une première approximation des différents sommets du squelette qui serviront de paramètres de synthèse au modèle. En chacun de ces sommets, on évaluera un rayon englobant le feuillage. Grâce à cette première approximation, on obtient une première modélisation 3D de notre plante.

#### Affinage des paramètres

Le but de cette dernière partie est d'affiner la modélisation 3D de la plante. Pour cela, nous comparons l'extraction de la plante de l'image originale (Section 3.1, page 4) avec l'image issue de la projection perspective du modèle obtenu (Figure 10, page 7).

Les paramètres de synthèse deviennent alors les paramètres à faire varier lors de la minimisation. Dans un but de réalisme, nous avons vu précédemment que la modélisation du pied en 3D comporte des parties aléatoires. Ces dernières sont ignorées dans cette étape pour que l'optimisation puisse se faire correctement.

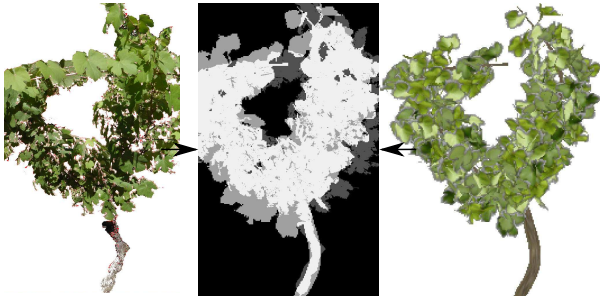


FIGURE 10 – Comparaison de l’image originale et de l’image reprojétée après modélisation 3D. À gauche, l’image originale. À droite, l’image resynthétisée de la plante via notre méthode. Au milieu la superposition des deux. Les pixels blancs signifient qu’il existe du feuillage ou du branchage sur l’image originale et sur l’image resynthétisée.

Nous devons maintenant trouver une métrique permettant de comparer deux images pouvant être vues comme deux images binaires (si feuillage ou branchage=1, sinon=0). Pour cela, on calcule une troisième image binaire  $\mathcal{B}$  correspondant à la valeur absolue de la différence entre les deux premières images binaires. Ainsi, tous les pixels ayant pour valeur 1 correspondent aux endroits où les pixels de l’image originale appartenant à la plante et ceux de l’image resynthétisée appartenant à la plante reconstruite ne se superposent pas. Ce sont ces pixels que l’on cherche à minimiser.

Pour le moment, nous proposons une méthode de minimisation qui, à chaque itération, modifie les paramètres. Le modèle génératif crée alors la modélisation 3D de la plante correspondant et la reprojette. Ensuite, après un recalage entre l’image originale et l’image reprojétée, on calcule l’image binaire des différences  $\mathcal{B}$ . L’erreur est alors le nombre de pixels de l’image  $\mathcal{B}$  ayant pour valeur 1.

Cependant, cette minimisation n’affine que peu la qualité de la modélisation 3D. Cette dernière partie reste encore à améliorer lors nos travaux futurs.

## 4 Résultats

Nous avons testé notre méthode sur les images de vigne dans les conditions d’acquisitions décrites précédemment (Section 3.1, page 4). Nous obtenons des résultats satisfaisants (Figure 11, 12 et 13, pages 7 et 8). L’erreur de projection est calculée comme étant le rapport entre les pixels de  $\mathcal{B}$  ayant pour valeur 1 et ceux ayant pour valeur 0 ou 1. Sur l’ensemble de nos expérimentations, cette valeur est en moyenne de 11.1%.

## 5 Conclusions et perspectives

La méthode présentée ici permet d’obtenir des reconstructions satisfaisantes dans le cas de modélisations de pieds de vigne. Cependant, le modèle génératif peut être encore amélioré ainsi que la partie visant à affiner le modèle en minimisant l’erreur entre l’image originale et l’image



FIGURE 11 – Exemple de modélisation d’un pied de vigne à partir d’une image. À gauche, l’image originale. À droite, l’image resynthétisée après reconstruction 3D de la plante puis reprojektion.



FIGURE 12 – À gauche, l’image originale avant rectification métrique. À droite, la même image sur laquelle on a surimprimé les pieds de vigne modélisés puis reprojétés.

age resynthétisée. En effet, une métrique plus efficace pourrait être choisie en se basant sur les attracteurs utilisés lors de la partie d’initialisation. Le modèle génératif doit être suffisamment contraint pour minimiser le nombre de paramètres de synthèse et suffisamment souple, pour obtenir une modélisation fidèle de la plante.

Dans le futur, notre but serait d’adapter notre modèle à d’autres types de plantes. Il est également envisagé de construire un modèle multi-échelle avec des paramètres de synthèse à différentes échelles. Ainsi, nous aimerions modéliser des parcelles de vigne entièrement en ajoutant éventuellement les grappes de raisin comme nous les avons modélisées dans des travaux précédents [GCMG09] [GM\*09].

## 6 Remerciements

Ce travail a été réalisé dans le cadre du projet VINNEO, financé par la République Française, l’Europe, les conseils régionaux du Languedoc-Roussillon et de Midi-Pyrénées, le conseil général du Tarn, et OSEO.

## Références

[BCPG10] BOUDON F., COKELAER T., PRADAL C., GODIN C. : L-py, an open l-systems frame-





FIGURE 13 – Autre exemple de modélisation de pieds de vigne à partir d'une image. A gauche, l'image originale. A droite, l'image resynthétisée après reconstruction 3D des plantes puis reprojection.

- work in python. *6th International Workshop on Functional-Structural Plant Models* (2010), 116–119.
- [BMG06] BOUDON F., MEYER A., GODIN C. : Survey on computer representations of trees for realistic and efficient rendering. *Research report*, RR-LIRIS-2006-003 (2006).
- [DL05] DEUSSEN O., LINTERMANN B. : *Digital Design of Nature : Computer Generated Plants and Organics*. Springer-Verlag, 2005.
- [GCMG09] GURDJOS P., CHARVILLAT V., MORIN G., GUÉNARD J. : Multiple view reconstruction of a quadric of revolution from its occluding contours. *ACCV* (2009), 1–12.
- [GGM\*10] GUÉNARD J., GIRON C., MORIN G., GURDJOS P., BOUDON F., CHARVILLAT V. : Modélisation de vignes à partir d'une séquence d'images. *AFIG* (2010).
- [GM\*09] GUÉNARD J., MORIN G., GURDJOS P., CHARVILLAT V. : De la reconstruction de quadriques de révolution à partir d'images à la complémentation d'objets naturels. *AFIG* (2009).
- [GS05] GODIN C., SINOQUET H. : Functional-structural plant modelling. *New Phytologist* (2005), 705–708.
- [GsK05] GURDJOS P., SO KWEON I. : Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration. *PAMI* (2005).
- [HZ00] HARTLEY R., ZISSERMAN A. : *Multiple view geometry in computer vision*. Cambridge University Press, 2000.
- [MBP\*08] MAIRAL J., BACH F., PONCE J., SAPIRO G., ZISSERMAN A. : Discriminative learned dictionaries for local image analysis. *CVPR* (2008), 1–8.
- [NFD07] NEUBERT B., FRANKEN T., DEUSSEN O. : Approximate image-based tree-modeling using particle flows. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2007)* (2007).
- [NSH08] NAIR V., SUSSKIND J., HINTON G. E. : Analysis-by-synthesis by learning to invert generative black boxes. *ICANN* (2008), 971–981.
- [PHL\*09] PALUBICKI W., HOREL K., LONGAY S., RUNIONS A., LANE B., MÈCH R., PRUSINKIEWICZ P. : Self-organizing tree models for image synthesis. *SIGGRAPH* (2009), 1–10.
- [PL90] PRUSINKIEWICZ P., LINDENMAYER A. : *The algorithmic beauty of plants*. Springer Verlag, 1990.
- [PT97] PIEGL L., TILLER W. : *The NURBS book (2nd ed.)*. Springer-Verlag New York, Inc., 1997.
- [QTZ\*06] QUAN L., TAN P., ZENG G., YUAN L., WANG J., KANG S. B. : Image-based plant modeling. *ACM Trans. Graph.* (2006), 599–604.
- [QWTY07] QUAN L., WANG J., TAN P., YUAN L. : Image-based modeling by joint segmentation. *IJCV* (2007), 135–150.
- [RMMD04] RECHE-MARTINEZ A., MARTIN I., DRETAKIS G. : Volumetric reconstruction and interactive rendering of trees from photographs. *ACM Trans. Graph.* (2004), 720–727.
- [TCYZ03] TU Z., CHEN X., YUILLE A. L., ZHU S.-C. : Image parsing : Unifying segmentation, detection, and recognition. *PAMI* (2003).
- [TFX\*08] TAN P., FANG T., XIAO J., ZHAO P., QUAN L. : Single image tree modeling. *SIGGRAPH* (2008), 1–7.
- [TK91] TOMASI C., KANADE T. : Detection and tracking of point features. *IJCV* (1991).
- [YK06] YUILLE A. L., KERSTEN D. : Vision as bayesian inference : Analysis by synthesis? *Trends in Cognitive Sciences In Probabilistic models of cognition* (2006).
- [ZZZ06] ZENG J., ZHANG Y., ZHAN S. : 3d tree models reconstruction from a single image. *ISDA* (2006), 445–450.