

Une approche pour l'analyse et exploitation des contraintes géométriques en modélisation 3D

Marta Wilczkowiak, Peter Sturm, Edmond Boyer

► **To cite this version:**

Marta Wilczkowiak, Peter Sturm, Edmond Boyer. Une approche pour l'analyse et exploitation des contraintes géométriques en modélisation 3D. Journées ORASIS, May 2003, Gérardmer, France. pp.77-86, 2003. <inria-00525638>

HAL Id: inria-00525638

<https://hal.inria.fr/inria-00525638>

Submitted on 26 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.

Une approche pour l'analyse et exploitation des contraintes géométriques en modélisation 3D

3D modelling using geometric constraints – a practical method based on the analysis of ambiguous solutions

Marta Wilczkowiak * Peter Sturm Edmond Boyer

Movi-Gravir-Inria Rhône-Alpes, 655 Avenue de l'Europe, 38330 Montbonnot, France

Prenom.Nom@inrialpes.fr

Résumé

Nous présentons une approche interactive pour la reconstruction de scènes tridimensionnelles à partir d'images et de contraintes géométriques. L'objectif est de reconstruire des modèles fiables à partir d'une, ou de très peu d'images, de calculer des positions de points occultés dans les images considérées, ou simplement d'imposer certaines propriétés au modèle. Une scène est modélisée par des points, des droites et des plans, toutes ces primitives étant représentées par des vecteurs de coordonnées homogènes. Cette représentation permet de formuler la plupart des contraintes géométriques, comme incidence, coplanarité, orthogonalité ou projection sous la forme d'équations linéaires. Contrairement aux approches existantes, notre méthode est capable non seulement de détecter si un système est sous- ou sur-contraint, mais aussi de séparer les variables du système en variables calculables et non-calculables, tout en permettant de contraindre plusieurs primitives à la fois par la même contrainte. L'idée est d'exploiter de manière optimale l'ensemble de l'information disponible. Des résultats sur des images réelles illustrent son fonctionnement.

Mots Clés

reconstruction 3D, contraintes géométriques, méthodes linéaires.

Abstract

We present a system for interactive scene modelling using multiple images and geometric constraints on the scene. The scene is reconstructed as a set of points, lines and planes. All primitives are represented using homogeneous coordinates, which leads to linear or bilinear expressions of a number of geometric constraints including: incidence, coplanarity, orthogonality, parallelism, projection (3D point/line onto image point/line). The basic practical reconstruction approach is iterative. At each iteration, as many primitives as possible are reconstructed from previously reconstructed ones and the geometric constraints. In practice, the solved systems are often underconstrained, either because sets of primitives are not related by any constraint or because the available constraints are insufficient or ill-conditioned. We thus incorporate an analysis that aims at detecting and solving these problems, without which meaningless results would be obtained. Applications to scene modelling are discussed and preliminary results on real data are presented.

Keywords

3D reconstruction, geometric constraints, linear methods.

1 Introduction

La construction de modèles 3D à partir d'images est un problème complexe, du fait de la diversité des scènes observées et du bruit induit par l'acquisition d'images. Pour rendre le processus de modélisation plus robuste, et donc les résultats plus réalistes, des contraintes géométriques sont très souvent incorporées au processus.

Différentes approches ont été proposées pour exploiter des contraintes géométriques dans les processus de modélisation à partir d'images. Elles peuvent être classées en fonction des primitives considérées pour modéliser la scène. Un premier groupe d'approches utilise des primitives de base, telles que points, droites et plans [14, 4, 16, 12, 11]. Un deuxième groupe utilise des primitives plus évoluées, comme les prismes, les cubes etc.[5, 3], ou même des modèles plus complexes [10]. Les approches du premier groupe nécessitent une interaction importante avec l'utilisateur, ce qui est compensé par une grande flexibilité dans la définition des contraintes possibles sur le modèle. Pour illustrer l'intérêt de cette flexibilité, considérons par exemple les bâtiments anciens. Les angles ne sont pas exactement droits et les murs ne sont pas parfaitement parallèles, forcer ces contraintes au travers de primitives complexes telle qu'un cube risque alors de déformer d'autres parties du modèle final. Les deux groupes d'approches présentent par contre des intérêts communs. En effet, l'utilisation de contraintes géométriques permet non seulement d'améliorer les résultats, mais aussi de reconstruire à partir de très peu d'images, avec peu de redondances d'information entre les images. Cela permet aussi de reconstruire des points de la scène qui sont occultés dans les images.

L'approche que nous proposons est basée sur un formalisme similaire à celui utilisé dans [12] et [11]. La scène est représentée à l'aide de points, droites et plans exprimés en coordonnées homogènes. Les contraintes géométriques classiques telles que projection, incidence, parallélisme et orthogonalité sont alors linéaires en les coordonnées des

*M. Wilczkowiak est financée par le projet IST-1999-10756, VISIRE.

primitives, et nous utilisons un algorithme itératif pour effectuer la reconstruction. À la différence de [12], notre méthode permet de contraindre plusieurs primitives simultanément.

À chaque itération, l'ensemble des contraintes est rassemblé pour former un système linéaire en les coordonnées de toutes les primitives de la scène. En pratique, ce système est souvent sous- ou sur-contraint (ou sous-contraint *et* sur-contraint en même temps pour différents sous-ensembles d'inconnues). Nous nous intéressons à la détection et la résolution de ce type de problèmes au travers d'une analyse du système linéaire formé par les contraintes. Dans cet article nous nous focalisons sur le problème de la reconstruction. Bien entendu les contraintes géométriques utilisées pour la reconstruction peuvent aussi servir à calibrer les caméras [4, 19, 17, 18].

2 Préliminaires

2.1 Notations

Les vecteurs et matrices, ainsi que les objets qu'ils représentent, sont écrits en caractères gras, par exemple un vecteur \mathbf{X} représente un point \mathbf{X} . Pour une matrice \mathbf{A} , les notations suivantes sont utilisées : \mathbf{A}^i pour une colonne, \mathbf{A}_i pour une ligne et \mathbf{A}^\dagger pour sa pseudo-inverse [1, 8].

$[\mathbf{x}]_\times$ représente la matrice antisymétrique associée au produit vectoriel : $[\mathbf{x}]_\times \mathbf{y} = \mathbf{x} \times \mathbf{y}$.

Une caméra est décrite par le modèle à *sténopé*. La transformation projective correspondante est représentée par une matrice $\mathbf{P}_{3 \times 4}$, telle que l'image \mathbf{x} du point \mathbf{X} est $\mathbf{x} \sim \mathbf{P}\mathbf{X}$ (où \sim représente l'égalité entre deux vecteurs des coordonnées projectives).

2.2 Entités géométriques manipulées

La scène est modélisée par des points, des droites et des plans. Comme remarqué dans [11], chacune de ces entités peut être décrite par un vecteur de coordonnées consistant en une partie homogène et une partie euclidienne : $\mathbf{V}^\top = (\mathbf{V}_0^\top \ \mathbf{V}_h^\top)$ (voir le tableau 1). Ce vecteur est défini à un facteur d'échelle près. Ce facteur est fixé de façon à ce que $\|\mathbf{V}_h\| = 1$. En conséquence, la distance d'un objet à l'origine est égal à $d = \|\mathbf{V}_0\|$.

Les coordonnées de Plücker d'une droite. Une représentation d'une droite satisfaisant les conditions énoncées précédemment (soit un vecteur de coordonnées homogènes) est donnée par les coordonnées de Plücker. Il s'agit d'un vecteur de dimension 6 : $\mathbf{L}^\top = (\mathbf{L}_h^\top \ \mathbf{L}_0^\top)$ vérifiant la condition de Plücker : $\mathbf{L}_h^\top \mathbf{L}_0 = 0$. Le vecteur \mathbf{L}_h correspond à la direction de la droite et le vecteur \mathbf{L}_0 à la normale au plan contenant la droite et l'origine. La condition de Plücker impose l'orthogonalité de ces deux vecteurs.

Les relations entre les entités décrites ci-dessus s'expriment de manière très élégante à l'aide de l'algèbre de Grassman-Cayley. Dans cet article, nous nous focalisons sur les contraintes linéaires dérivées à l'aide de cette

algèbre. Le lecteur intéressé trouvera une introduction théorique dans [6] par exemple. L'algèbre de Grassman-Cayley introduit deux opérateurs, *meet* Δ et *join* ∇ . L'opérateur *meet* caractérise l'intersection de deux entités et l'opérateur *join* caractérise l'espace linéaire minimal contenant deux entités. Par exemple, un point peut être défini comme le *meet* de trois plans, et un plan peut être défini comme le *join* de trois points. Pour faciliter la formulation des relations entre entités, nous utilisons une représentation matricielle. Le tableau 1 précise quelles sont ces matrices, leurs interprétations géométriques peuvent être trouvées dans [11].

2.3 Contraintes

Nous nous focalisons sur les contraintes linéaires et bilinéaires entre les entités décrites ci-dessus. Les relations de projection, d'incidence, d'orthogonalité et de parallélisme entre deux entités sont bilinéaires en les coordonnées de chacune des entités. À cela s'ajoutent des contraintes linéaires issues, par exemple, de relations de symétrie ou du fait que quatre points forment un parallélogramme.

Les contraintes bilinéaires Les tableaux 2,3,4 et 5 donnent les équations engendrées par une relation de projection, d'incidence (nous considérons l'égalité entre deux entités comme une relation d'incidence), d'orthogonalité et de parallélisme entre deux entités. Ces équations sont bilinéaires en les coordonnées des deux entités et deviennent linéaires en les coordonnées d'une des entités objet lorsque celles de la deuxième sont connues. Pour chaque contrainte le nombre de degrés de liberté (ddl) fixé par cette contrainte est précisé.

projection \mathbf{P}			
(ddl)	point \mathbf{X}	(ddl)	droite \mathbf{L}
(2)	$\mathbf{P}\mathbf{X} \sim \mathbf{x}$	(2)	$\begin{bmatrix} \mathbf{P}_2 \Delta \mathbf{P}_3 \\ \mathbf{P}_3 \Delta \mathbf{P}_1 \\ \mathbf{P}_1 \Delta \mathbf{P}_2 \end{bmatrix}_{3 \times 6} \mathbf{L} \sim \mathbf{l}$

Table 2: Les contraintes sur les points et droites 3D engendrées par des projections images. Les \mathbf{P}_i^\top sont les lignes de la matrice de projection \mathbf{P} pour les points et les $\mathbf{P}_i \Delta \mathbf{P}_j$ sont des coordonnées de Plücker des droites d'intersections de plans représentées par P_i et P_j .

	l'inconnue	
	(ddl) droite \mathbf{L}	(ddl) plan \mathbf{A}
donnéee \downarrow		
droite \mathbf{M}	(2) $[\mathbf{M}_h]_\times \mathbf{L}_h = \mathbf{0}$ ou $\mathbf{L}_h := \mathbf{M}_h$	(1) $\mathbf{M}_h^\top \mathbf{A}_h = 0$
plan \mathbf{B}	(1) $\mathbf{B}_h^\top \mathbf{L}_h = 0$	(2) $[\mathbf{B}_h]_\times \mathbf{A}_h = \mathbf{0}$ ou $\mathbf{A}_h := \mathbf{B}_h$

Table 4: Les contraintes engendrées par des relations de parallélisme.

Les relations d'égalité et de parallélisme (égalité des directions) peuvent être appliquées de deux façons différentes.

objet	ddl	vecteur représentation	matrice associée
point	3	$\mathbf{X}^\top = (x, y, z, t) = (\mathbf{X}_0^\top \mathbf{X}_h)$	$\Pi(\mathbf{X}) = \begin{bmatrix} X_h * \mathbf{I}_{3 \times 3} & [\mathbf{X}_0]^\times \\ -\mathbf{X}_0^\top & \mathbf{0}^\top \end{bmatrix}_{4 \times 6}$
droite	4	$\mathbf{L}^\top = (l_0, l_1, l_2, l_3, l_4, l_5) = (\mathbf{L}_h^\top \mathbf{L}_0^\top)$	$\Gamma(\mathbf{L}) = \begin{bmatrix} -[\mathbf{L}_h]^\times & -\mathbf{L}_0 \\ \mathbf{L}_0^\top & 0 \end{bmatrix}_{4 \times 4}$ $\bar{\Gamma}(\mathbf{L}) = \Gamma(\bar{\mathbf{L}}) = \begin{bmatrix} -[\mathbf{L}_0]^\times & -\mathbf{L}_h \\ \mathbf{L}_h^\top & 0 \end{bmatrix}_{4 \times 4}$
plan	3	$\mathbf{A}^\top = (a, b, c, d) = (\mathbf{A}_h^\top \mathbf{A}_0)$	$\bar{\Pi}(\mathbf{A}) = \begin{bmatrix} -[\mathbf{A}_h]^\times & \mathbf{A}_0 * \mathbf{I}_{3 \times 3} \\ -\mathbf{A}_h^\top & \mathbf{0}^\top \end{bmatrix}_{4 \times 6}$

Table 1: Représentation des points, droites et plans en 3D. Pour chaque objet le nombre de degrés de liberté (ddl) est donné.

incidence	l'inconnue			
donnée ↓	(ddl) point \mathbf{X}	(ddl) droite \mathbf{L}	(ddl) plan \mathbf{A}	
point \mathbf{Y}	(3) $\Pi^\top(\mathbf{Y})\mathbf{X} = \mathbf{0}$	(2) $\bar{\Pi}(\mathbf{Y})\mathbf{L} = \mathbf{0}$	(1) $\mathbf{Y}^\top \mathbf{A} = \mathbf{0}$	
ligne \mathbf{M}	(2) $\Gamma(\mathbf{M})\mathbf{X} = \mathbf{0}$	(4) $\bar{\Gamma}(\mathbf{M})\Gamma(\mathbf{L}) = \mathbf{0}$	(2) $\bar{\Gamma}(\mathbf{M})\mathbf{A} = \mathbf{0}$	
plan \mathbf{B}	(1) $\mathbf{B}^\top \mathbf{X} = \mathbf{0}$	(2) $\Pi(\mathbf{B})\mathbf{L} = \mathbf{0}$	(3) $\Pi^\top(\mathbf{B})\mathbf{A} = \mathbf{0}$	

Table 3: Les contraintes engendrées par des relations d'incidence.

\perp	l'inconnue	
donnée ↓	(ddl) droite \mathbf{L}	(ddl) plan \mathbf{A}
droite \mathbf{M}	(1) $\mathbf{M}_h^\top \mathbf{L}_h = 0$	(2) $[\mathbf{M}_h]^\times \mathbf{A}_h = \mathbf{0}$ ou $\mathbf{A}_h := \mathbf{M}_h$
plan \mathbf{B}	(2) $[\mathbf{B}_h]^\times \mathbf{L}_h = \mathbf{0}$ ou $\mathbf{L}_h := \mathbf{B}_h$	(1) $\mathbf{B}_h^\top \mathbf{A}_h = 0$

Table 5: Les contraintes engendrées par des relations d'orthogonalité.

En premier lieu, les équations engendrées par ces relations peuvent être ajoutées au système global contenant toutes les équations. En deuxième lieu, les entités qui sont égales peuvent être représentées par un seul jeu de variables. La première solution permet de traiter toutes les contraintes de manière équivalente. Cependant, la solution obtenue ne respectera pas exactement les contraintes. La deuxième solution permet d'imposer de manière exacte certaines contraintes. Un moyen général de respecter certaines contraintes *exactement* est présenté dans le paragraphe 4.4.

Les contraintes linéaires. Les contraintes telles que l'appartenance de quatre points à un parallélogramme ou les relations de symétrie ne peuvent pas être incorporées dans le formalisme de l'algèbre de Grassman-Cayley. Cependant, elles sont très simples à inclure dans le système global et se révèlent très efficaces en pratique. D'une manière générale toutes les contraintes du type $\sum_{i=1}^n \alpha_i \mathbf{X}_i = \mathbf{0}$ peuvent s'exprimer de la façon suivante :

$$\begin{bmatrix} \alpha_1 * \mathbf{I} & \alpha_2 * \mathbf{I} & \cdots & \alpha_n * \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_n \end{bmatrix} = \mathbf{0}, \quad (1)$$

Par exemple, le fait que quatre points $\mathbf{X}_1 \cdots \mathbf{X}_4$ forment un parallélogramme peut s'écrire sous la forme (1), avec $n = 4$ et $\alpha_1 = \alpha_3 = -\alpha_2 = -\alpha_4$.

3 Le principe de la reconstruction

3.1 Description du système

Il s'agit de reconstruire des points, des droites et des plans à l'aide de contraintes géométriques. Nous nous focalisons sur les contraintes linéaires portant sur une ou plusieurs entités géométriques. Les contraintes et les entités sont introduites par l'utilisateur de manière interactive. La résolution est ensuite automatique : les entités pour lesquelles un nombre suffisant de contraintes sont disponibles seront reconstruites, les autres seront détectées. En général, seules quelques entités peuvent être reconstruites directement à l'initialisation en une itération, les autres nécessitent plusieurs itérations. Considérons par exemple un point \mathbf{X}_0 défini par l'intersection de deux droites \mathbf{L}_0 et \mathbf{L}_1 , chacune des droites étant définie par deux points. Sans contraintes additionnelles, \mathbf{X}_0 ne peut être calculé qu'après l'estimation préalable des droites \mathbf{L}_0 et \mathbf{L}_1 .

De manière similaire, une contrainte d'orthogonalité entre deux droites influence la solution uniquement (dans le système linéaire résolu) lorsqu'au moins une des droites a été calculée. C'est pourquoi il est nécessaire de procéder itérativement. À chaque itération, toutes les entités pour lesquelles existe un nombre suffisant de contraintes linéaires sont déterminées. La scène est modélisée par un graphe avec pour nœuds les objets et pour arêtes les contraintes. Par exemple, quatre points coplanaires sont représentés par cinq nœuds (4 points et 1 plan) et quatre arêtes (les relations d'incidence entre les points et le plan). Le schéma d'un algorithme basé sur ce principe est donné dans la figure 1.

Après chaque itération, une étape de recalibrage de la caméra peut éventuellement être ajoutée comme suggéré dans [12].

Pour que cet algorithme fonctionne, deux problèmes doivent être résolus : définir la condition d'arrêt ainsi

repete

1. $N := \sum_{i=1}^n \text{nb_coords}(\text{obj}_i)$.

2. initialiser un système linéaire vide :

$$\mathbf{A}_{0 \times N} \mathbf{X}_{N \times 1} = \mathbf{B}_{0 \times 1}.$$

3. trouver la fonction d'indexation (bijection)

$$F : id \rightarrow (i, j); id \in 1 \dots N$$

où id est un index dans le vecteur des inconnues \mathbf{X} et (i, j) correspond à la j -ème coordonnée du i -ème objet.

4. **pour chaque** contrainte c^k :

- calculer

$$(A_{m_k \times N}^k, B_{m_k \times 1}^k) := \text{contrainte_equations}(c^k.type, c^k.objects);$$

- ajouter au système global d'équations :

$$\mathbf{A} := \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^k \end{bmatrix}, \quad \mathbf{B}^k := \begin{bmatrix} \mathbf{B} \\ \mathbf{B}^k \end{bmatrix}.$$

5. résoudre le système $\mathbf{A}\mathbf{X} = \mathbf{B}$.

6. **pour** $id = 1 \dots N$

si (variable_calculée(id)):

- $(i, j) := F(id)$;
- $(\text{obj}_j.coords(j)) := \mathbf{X}(id)$.

jusqu'à "condition d'arrêt".

Figure 1: L'algorithme de reconstruction.

qu'un critère pour détecter les variables suffisamment bien définies.

La condition d'arrêt dépend de l'application envisagée. L'algorithme peut s'arrêter quand toutes les entités bien définies ont été calculées et/ou quand toutes les contraintes sont respectées avec une exactitude prédéfinie. Le deuxième problème nécessite quant à lui une solution explicite. En général, et surtout dans les premières itérations, l'information donnée à l'algorithme est insuffisante pour reconstruire l'ensemble des objets constituant la scène. En d'autres termes, la matrice \mathbf{A} est de rang insuffisant et la solution \mathbf{X} ne peut être calculée directement. Par contre, certaines inconnues sont suffisamment contraintes pour être calculées. Une première étape pour séparer les inconnues qui sont calculables de celles qui ne le sont pas est trivial : le système global contenant l'ensemble des équations $\mathbf{A}\mathbf{X} = \mathbf{B}$ doit être divisé en plusieurs sous-systèmes indépendants $\mathbf{A}_i \mathbf{X}_i = \mathbf{B}_i$ de la façon, que tous les variables dans un sous-ensemble sont liées par des

équations. Cela permet d'obtenir des sous-systèmes dépendants de moins des variables, et de séparer ainsi des parties du modèle qui sont parfois suffisamment contraintes. Malheureusement, les sous-systèmes construits de la sorte restent fréquemment sous-contraints pour certaines variables. En conséquence, et pour pouvoir utiliser pleinement l'information donnée, nous introduisons une méthode qui permet de détecter et de calculer les variables indépendantes et donc déterminables et de marquer les variables non-déterminables.

3.2 Variables solvables

Notre approche est basée sur l'analyse du noyau de l'application linéaire associée à la matrice \mathbf{A} . Soit $\Phi(\mathbf{A})$ le noyau de \mathbf{A} et $\{\mathbf{v}_1, \dots, \mathbf{v}_j\}$ une base de ce noyau. Soit \mathbf{X}_0 un vecteur satisfaisant le système d'équations $\mathbf{A}\mathbf{X} = \mathbf{B}$. Par définition du noyau, chaque vecteur \mathbf{X} satisfaisant le système d'équations peut s'écrire sous la forme :

$$\mathbf{X} = \mathbf{X}_0 + \sum_{i=1}^j \lambda_i \mathbf{v}_i, \lambda_i \in \mathbb{R}.$$

La solution $\mathbf{X}(k)$ pour le k -ième élément de \mathbf{X} est unique, si et seulement si : $\forall \lambda_i \in \mathbb{R}, \mathbf{X}(k) = \mathbf{X}_0(k)$. Cela implique que : $\forall \lambda_i \in \mathbb{R}, \sum_{i=1}^j \lambda_i \mathbf{v}_i(k) = 0$, ce qui est équivalent à :

$$\forall i \in \{1..j\}, \mathbf{v}_i(k) = 0. \quad (2)$$

En d'autres termes, une variable correspondant au k -ième élément de \mathbf{X} est déterminée de manière unique à l'aide de \mathbf{A} lorsque l'axe de \mathbb{R}^n associée à cette variable est orthogonal au noyau de \mathbf{A} , et sa valeur est alors donnée par $v_k = \mathbf{X}_0(k)$.

Pour déterminer le noyau d'une matrice et résoudre le système d'équations plusieurs méthodes sont possibles. Nous utilisons la décomposition SVD (décomposition en valeurs singulières) de \mathbf{A} : $\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T$, où les matrices \mathbf{U} et \mathbf{V} sont orthogonales, et \mathbf{W} est diagonale, avec pour éléments les valeurs singulières de la matrice \mathbf{A} en ordre décroissant sur la diagonale. Le noyau de la matrice \mathbf{A} , $\Phi(\mathbf{A})$ est engendré par les vecteurs colonne $\mathbf{V}^i, i \in r + 1..n$ de la matrice \mathbf{V} correspondant aux $(n - r)$ valeurs singulières $w_i = 0$. Le rang de la matrice (\mathbf{A}) est r .

Toutes les solutions \mathbf{X} du système peuvent alors être écrites sous la forme :

$$\begin{aligned} \mathbf{X} &= \mathbf{X}_0 + \sum_{i=r+1}^n \lambda_i \mathbf{V}^i, & \lambda_i &\in \mathbb{R} \\ &= \mathbf{A}^\dagger \mathbf{B} + \sum_{i=r+1}^n \lambda_i \mathbf{V}^i, & \lambda_i &\in \mathbb{R} \end{aligned}$$

où $\mathbf{X}_0 = \mathbf{A}^\dagger \mathbf{B}$ est un vecteur de longueur minimal satisfaisant l'équation $\mathbf{A}\mathbf{X} = \mathbf{B}$.

4 Détails d'algorithme

4.1 Détermination du rang d'une matrice

Un des problèmes à résoudre pour la reconstruction est la détermination du noyau d'une matrice, et donc de son rang. En raison des erreurs de données et des erreurs numériques,

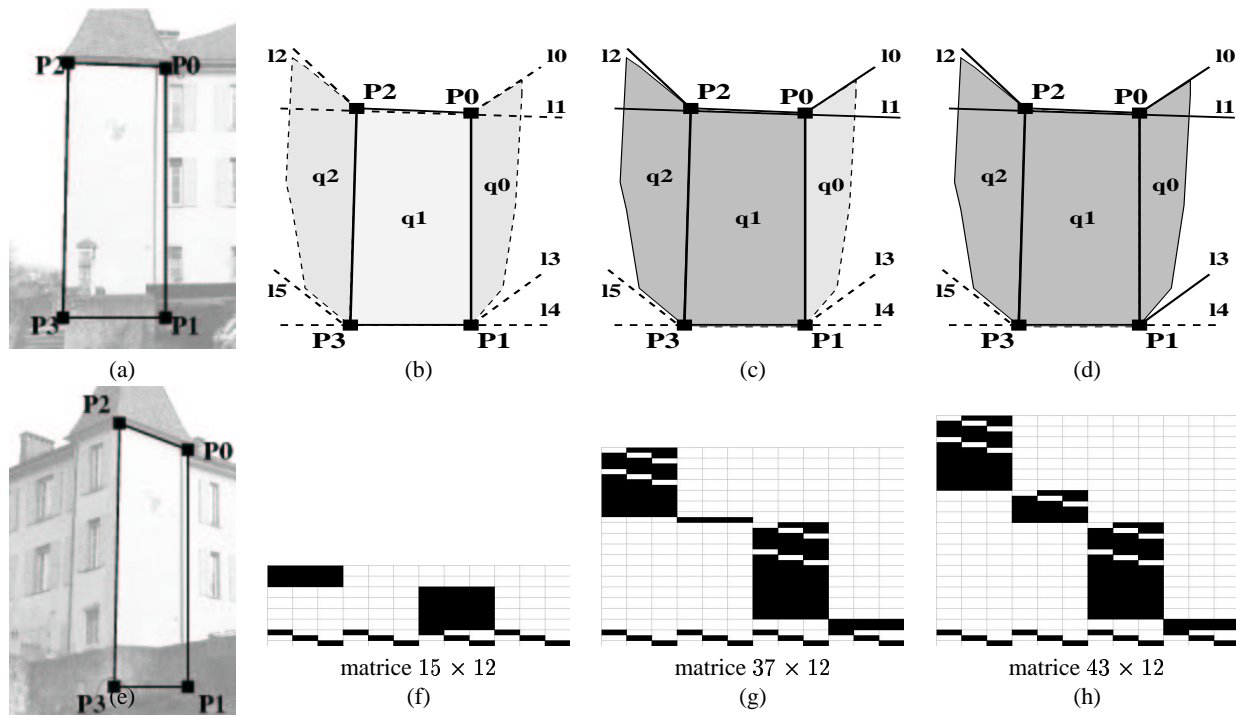


Figure 2: Exemple de reconstruction d’une partie du modèle ; première colonne : 2 images originales, où les points d’intérêt sont marqués; les points P1, P3 ne sont visibles dans aucune image; 2, 3, 4-ème colonne : les contraintes géométriques portant sur quatre points d’intérêt et la forme des matrices contenant toutes les équations engendrées par ces contraintes et les contraintes de projection. Dans la première ligne, les objets qui peuvent influencer les points P1 – P3 (leurs coordonnées sont connues à cette étape) sont marqués avec une ligne discontinue et couleur claire. Dans la deuxième ligne, les champs noirs correspondent aux éléments non-nuls de matrices.

les valeurs singulières w_i calculées en pratique ne sont jamais nulles. Une solution classique à ce problème est basée sur une analyse par perturbation de la matrice avec la méthode de Monte Carlo [13]. Cependant, cela nécessite plusieurs décompositions de la matrice. Comme la recherche du rang d’une matrice est effectuée à chaque itération, il est nécessaire d’utiliser des méthodes plus rapides. Précisément, toutes les valeurs w_k satisfaisants $w_k < tol = \max(size(\mathbf{A}))w_0\epsilon_0$ et $\frac{w_{k-1}-w_k}{w_k} < \epsilon_1$ sont considérées comme nulles. La deuxième condition repose sur l’hypothèse qu’en dépit du bruit, il existe une différence importante entre les w_i correspondant à des valeurs singulières nulles et les autres valeurs w_i .

L’efficacité de la méthode dépend du choix des seuils ϵ_0 et ϵ_1 . Si ϵ_0 est trop grand et ϵ_1 trop petit, il est possible que certaines variables bien définies soient considérées comme sous-contraintes. Dans la situation inverse certaines entités sous-contraintes seront considérées comme déterminables et influenceront les itérations suivantes. Nous utilisons les valeurs $\epsilon_0 = 10^{-4}$ et $\epsilon_1 = 10^{-2}$.

En pratique, si le système $\mathbf{AX} = \mathbf{B}$ est divisé en sous-systèmes indépendants, comme cela est décrit dans la section 3.1, normalement il n’y a pas d’ambiguïtés dans le choix des valeurs que doivent être considérées comme nulles et les valeurs de ϵ_0 et ϵ_1 données dans cette section ont été correctes pour toutes nos expérimentations.

4.2 Détection des variables déterminables

Décider si une variable est suffisamment bien contrainte pour être reconstruite nécessite de tester la condition (2). Comme dans la section précédente, les valeurs $v_i(k)$ calculées ne sont jamais nulles. Mais contrairement à la détermination du rang d’une matrice, un simple seuillage $v_i(k) < \epsilon$ donne de très bons résultats. Nous utilisons en pratique $\epsilon = 10^{-16}$.

4.3 Un exemple

Un exemple qui montre la capacité de notre méthode d’utiliser l’information de la façon optimale est montré sur la Figure 2. Grâce à l’analyse du noyau et séparation dans chaque itération des variables bien contraintes et de variables sous-contraintes à chaque étape on peut utiliser pleinement l’information disponible.

Dans cet exemple on cherche à reconstruire les points **P0-P3** montrés sur la Figure 2-(a) et 2-(e). Points **P0** et **P2** sont visibles dans plusieurs images. Des points **P1** et **P3** sont cachés par le mur. Des points **P0-P3** font partie de 3 planes et 6 lignes du modèle (voir Fig. 2-(b) – 2-(d)), qui sont liées aussi avec des autres primitives du modèle par les contraintes de incidence et du parallélisme. À part cela une contrainte que points **P0-P3** forment un parallélogramme a été imposée. Les colonnes 2-4 de la Figure 2 montrent l’état de la reconstruction dans la 1-ère, 2-ème et 4-ème itération.

iteration 0				iteration 1	
W	fi			W	fi
3.9	<i>-9.4e-17</i>	<i>1.9e-17</i>	<i>3.6e-18</i>	1.6e+02	<i>3.2e-17</i>
3.4	<i>1.5e-16</i>	<i>1.7e-17</i>	<i>-6.3e-17</i>	1.3e+02	<i>3.6e-17</i>
3.2	<i>-7.1e-17</i>	<i>2.5e-18</i>	<i>4.3e-17</i>	77	<i>2.8e-17</i>
2.7	-0.63	0.14	-0.29	77	0.71
2.3	-0.23	0.25	0.62	3.8	-0.046
1.5	-0.23	-0.65	0.18	3.2	-0.0041
1.3	<i>9.2e-17</i>	<i>-1.7e-17</i>	<i>3.8e-17</i>	1.7	<i>-3.2e-17</i>
1.2	<i>-8.3e-17</i>	<i>-2.7e-17</i>	<i>-1.7e-17</i>	1.6	<i>-2.1e-17</i>
0.77	<i>-4.1e-17</i>	<i>-3.0e-18</i>	<i>-3.3e-18</i>	1.4	<i>-7.3e-17</i>
2.1e-16	-0.63	0.14	-0.29	1.0	0.71
8.3e-17	-0.23	0.25	0.62	0.61	-0.046
3.2e-17	-0.23	-0.65	0.18	1.1e-15	-0.0041

Table 6: Les résultats numériques pour l'exemple de la Figure 2. Pour chaque itération la première colonne contient les valeurs singulières de la matrice \mathbf{A} des contraintes. Une ligne horizontale sépare les valeurs qui ont été classifiés comme nulles. Les colonnes suivantes contiennent des vecteurs formant le noyau de \mathbf{A} . Les valeurs classifiées comme nulles ont été marquées en italique. Les variables correspondant aux lignes avec des valeurs nulles uniquement ont été classifiées comme reconstruits. Ces valeurs correspondent aux coordonnées des points $\mathbf{P0}$ et $\mathbf{P2}$ de la Figure 2.

Des résultats numériques se trouvent dans le Tableau 6. Dans chaque étape de la reconstruction (Voir Section 3.1), tous les équations possibles sont ajoutées dans le système générale des équations. En suite ce système est partagé en sous-systèmes indépendants. Dans cet exemple, pour les points $\mathbf{P0-P3}$ on obtient un sous-système portant sur 12 variables, correspondants aux coordonnées $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ des points.

1-ère itération. La matrice $\mathbf{A}_{15 \times 12}$ contient seulement des contraintes engendrées par des projections et le parallélogramme. La forme de la matrice \mathbf{A} est montrée sur la Figure 2-(b). Seule analyse de cette forme indique, que les variables correspondants à coordonnées des points $\mathbf{P1}$ et $\mathbf{P3}$ ne peuvent pas être calculées (seulement 3 équations portent sur ces 6 variables). L'analyse du noyau de la matrice confirme ce résultat. Les valeurs singuliers de cette matrice sont détaillées dans la première colonne du tableau 6. Le choix des valeurs qui doivent être considérés comme nulles était très claire. Aux 3 valeurs nulles correspondent 3 vecteurs formant une base du noyau de la matrice \mathbf{A} . Aussi ici, il n'y a pas des ambiguïtés dans le choix quels éléments doivent être considérés comme nulles. La réponse du système est en d'accord avec l'intuition, que seulement variables correspondants aux coordonnées des points $\mathbf{P0}$ et $\mathbf{P2}$ sont bien contraints.

2eme itération. La matrice $\mathbf{A}_{37 \times 12}$ est augmentée par les équations engendrées par les plans $\mathbf{q1}$, $\mathbf{q2}$ et les droites $\mathbf{I0}$, $\mathbf{I1}$, $\mathbf{I2}$ qui étaient reconstruits à partir de contraintes qui lient ces objets avec des primitives de la reste du modèle calculés dans la première itération. La structure de cette matrice (Fig. 2-(c)) suggère, qu'il y a suffisamment

des contraintes pour calculer tous les variables du système. Cependant, une analyse des valeurs singulières de cette matrice (4-ème colonne du Tab. 6) indique, qu'une des valeurs singulières est nulle. Une analyse des contraintes portant sur les points $\mathbf{P0-P3}$ explique la situation: les contraintes que points $\mathbf{P0-P3}$ forment un parallélogramme, et qu'ils sont coplanaires (le plan $\mathbf{q1}$) sont redondants, alors toujours il n'y a pas suffisamment des contraintes pour calculer les points $\mathbf{P1}$ et $\mathbf{P3}$.

4-ème itération. Le plan $\mathbf{q0}$ et la droite $\mathbf{I3}$ sont reconstruits. Après que les équations correspondants sont ajoutées à la matrice \mathbf{A} , le rapport entre son le plus grand et le plus petit valeur singulière est égale à $\frac{160}{0.97} \sim 165$. En conséquence tous les points $\mathbf{P0-P3}$ sont classifiés comme reconstruits.

4.4 Forcer des contraintes

Il peut être intéressant, pour le calcul du modèle, d'imposer le respect exact de certaines contraintes plutôt que de les ajouter au système d'équations résolu par les moindres carrés. Cela n'est bien entendu possible que si les contraintes considérées comme exactes ne sont pas contradictoires et redondantes.

Soit un système de m équations, où r équations doivent être respectées *exactement*. Sans perte de généralité, les lignes des matrices \mathbf{A} et \mathbf{B} peuvent être permutées pour satisfaire :

$$\mathbf{A}_{m \times n} = \begin{bmatrix} \mathbf{A}_e & r \times n \\ \mathbf{A}'_{(m-r) \times n} \end{bmatrix}; \quad \mathbf{B}_{m \times n} = \begin{bmatrix} \mathbf{B}_e & r \times n \\ \mathbf{B}'_{(m-r) \times n} \end{bmatrix} \quad (3)$$

où \mathbf{A}_e et \mathbf{B}_e correspondent aux équations qui doivent être respectées exactement, et \mathbf{A}' \mathbf{B}' aux autres équations. Le problème peut alors être formulé comme la minimisation de la fonction :

$$f(\mathbf{X}) = \|\mathbf{A}'\mathbf{X} - \mathbf{B}'\|,$$

sous la contrainte linéaire $\mathbf{A}_e\mathbf{X} = \mathbf{B}_e$.

Ce problème peut être résolu à l'aide des méthodes d'optimisation classiques, par exemple les multiplicateurs de Lagrange [7]. Nous utilisons une autre méthode de substitution basée sur une décomposition SVD [1, 8].

Considérons le système $\mathbf{A}_e\mathbf{X} = \mathbf{B}_e$. Comme noté dans la section §3.2, l'ensemble des solutions peut s'exprimer à l'aide de la décomposition SVD de la matrice \mathbf{A}_e suivant :

$$\begin{aligned} \mathbf{X}_e &= \mathbf{X}_{0e} + \sum_{k=r+1}^n \lambda_k \mathbf{V}^k; & \lambda_k &\in \mathbb{R}; \\ \mathbf{X}_{0e} &= \mathbf{A}_e^\dagger \mathbf{B}_e; \end{aligned} \quad (4)$$

Tous les vecteurs \mathbf{X}_e respectent les équations 1.. r du système initial exactement. Maintenant, la résolution du système $\mathbf{A}\mathbf{X} = \mathbf{B}$ est réduite au calcul des coefficients λ_k tels que l'équation $\mathbf{A}'\mathbf{X}_e = \mathbf{B}'$ soit satisfaite de manière optimale (ici au sens de moindres carrés).

À l'aide de (4), le problème peut être reformulé suivant :

$$\begin{aligned} \mathbf{A}'\mathbf{X}_e = \mathbf{B}' &\Leftrightarrow \mathbf{A}'\mathbf{X}_e = \mathbf{B}' \\ &\Leftrightarrow \mathbf{A}'\mathbf{A}_e^\dagger\mathbf{B}_e + \mathbf{A}'\left(\sum_{k=r+1}^n \lambda_k \mathbf{V}^k\right) = \mathbf{B}' \\ &\Leftrightarrow \underbrace{\mathbf{A}'}_{\mathbf{A}''} \left[\begin{array}{c} \mathbf{V}^{r+1} \\ \dots \\ \mathbf{V}^n \end{array} \right] \left[\begin{array}{c} \lambda_{r+1} \\ \lambda_{r+2} \\ \vdots \\ \lambda_n \end{array} \right] = \underbrace{\mathbf{B}'}_{\mathbf{B}''} - \mathbf{A}'\mathbf{A}_e^\dagger\mathbf{B}_e. \end{aligned}$$

Ce qui est encore un problème de minimisation linéaire qui peut être facilement résolu. L'avantage de cette méthode par rapport aux multiplicateurs de Lagrange est que le système d'équations à résoudre est de moindre importance.

4.5 L'analyse d'incertitude

Pour valider les résultats et réduire l'influence des entités possédant une grande incertitude, il serait utile d'incorporer au système une analyse d'incertitude. Une analyse de propagation d'erreurs pour la représentation de la scène utilisée dans notre approche peut être trouvée dans [11]. Une fois l'incertitude d'une entité/contraainte évaluée, toutes les équations engendrées par cet objet/contraainte peuvent être pondérées pour favoriser l'information provenant d'entités bien définies.

5 Résultats

5.1 Reconstruction a partir de deux images

Une des applications de notre méthode est de construire des modèles à partir de quelques photos et d'imposer des contraintes géométriques sur le modèle. Dans cette expérience, nous avons pris deux photos d'un bâtiment de deux points de vues opposés. Les positions des caméras ont été calculées de manière approximative. La plupart des points du modèle ne sont visibles que dans une des images utilisées pour la modélisation. Pour pouvoir calculer leurs positions on a utilisé des contraintes d'appartenance à des parallélogramme et de coplanarité. En raison de l'imprécision sur les positions de caméras, le modèle obtenu avec les contraintes uniquement est très bruité (Voir la figure 3-(a)). Pour remédier à ce problème, nous avons imposé des contraintes de parallélisme et d'orthogonalité entre droites et plans dans les trois directions principales. La Figure 3 montre les images originales et le modèle à différents stades de la modélisation. À la première itération, les seuls entités déterminables sont les points visibles dans deux images et les points qui leurs sont liés par des contraintes d'appartenance à des parallélogrammes. À la deuxième et troisième itération, les points liés aux points déjà reconstruits par des contraintes de coplanarité et de colinéarité apparaissent dans le modèle. Ensuite, les contraintes de parallélisme et d'orthogonalité commencent à affecter les points incidents à des plans et des droites. À la quatrième itération, le modèle prend sa forme finale. À la dixième et dernière itération, le modèle est stable et visuellement correct.

5.2 Reconstruction du château de Montbonnot

La deuxième expérience a été réalisée à partir de huit photos d'un château (figures 4-(a)–4-(h)). Les caméras correspondantes ont été calibrées en utilisant les approches décrites dans [17, 18]. La construction du modèle a fait apparaître les problèmes suivants :

1. Il y a peu de redondance d'information entre les images ce qui a pour conséquence de rendre l'estimation des poses des caméras difficile.
2. Certains points importants pour la qualité du modèle final ne sont pas visibles dans les images et d'autres sont visibles dans une image uniquement. Une conséquence est que, par exemple, la profondeur du mur visible dans les images 4-(f) et 4-(g) est très mal définie par les contraintes de projection.
3. Peu de contraintes géométriques peuvent être utilisées pour compenser le manque d'information sur les projections des points. Un plan de la mairie montré sur la figure 4-(p) met en évidence que peu de murs peuvent être considérés comme parallèles ou orthogonaux. Toutes les arêtes verticales du château sont légèrement penchées au centre du bâtiment, ce qui permet pas de les définir comme parallèles. Il faut donc être prudent sur le choix des contraintes imposées. Nous utilisons les contraintes géométriques principalement pour estimer les points qui ne sont pas visibles dans un nombre suffisant d'images.

Les résultats de la modélisation sont montrés dans la figure 4. La troisième ligne de la figure montre les parties du château reconstruites dans les quatre premières itérations de la reconstruction. Il apparaît que de nouvelles entités sont reconstruites à chaque itération. Les images 4-(m)–4-(o) montrent le modèle final texturé.

La figure 4-(p) montre la reprojection des points du modèle sur le plan de la mairie. Les points qui sont entourés par les cercles ont été reconstruits à partir des contraintes géométriques et non par triangulation. La figure montre que les erreurs de reprojection pour ces points sont du même ordre de grandeur que celle des points triangulés, ce qui démontre la fiabilité et l'intérêt d'utiliser des contraintes géométriques. Le plan du château peut être aussi utilisé pour la modélisation, ce qui rend le processus plus rapide et plus exact.

6 Conclusion

Nous avons présenté une approche pour exploiter des contraintes géométriques et projectives pour la modélisation de scènes 3D à partir d'images. Grâce à une représentation homogène des primitives constituant la scène, nous avons pu formuler les contraintes d'incidence, de parallélisme, de coplanarité et de symétrie à l'aide d'équations linéaires en les coordonnées des entités considérées. Cela

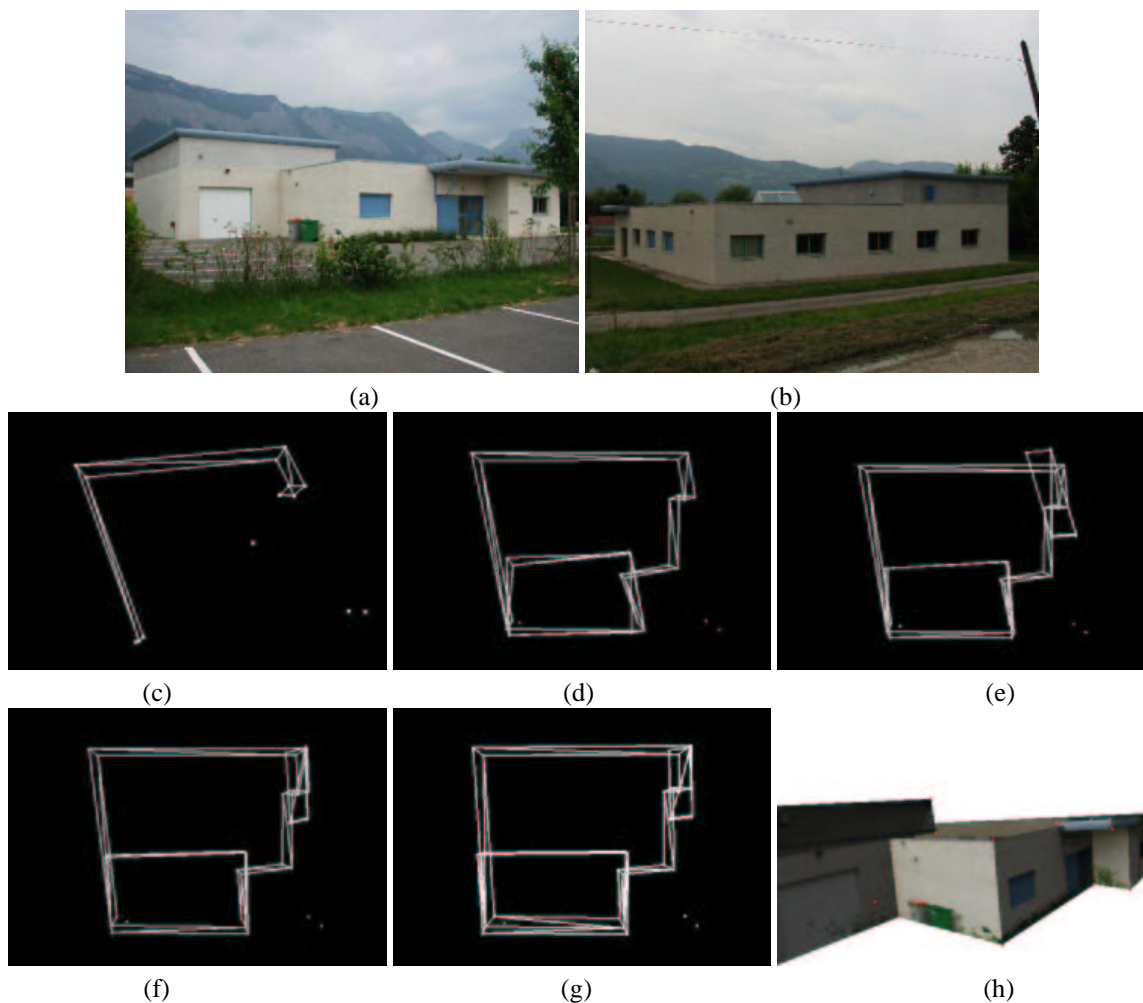


Figure 3: Reconstruction à partir de deux images; (a),(b) Les photos originales; (c) Les points reconstruits à partir des contraintes de projection et d'appartenance à des parallélogrammes (d),(e) Les objets définis par les contraintes de coplanarité et de colinéarité apparaissent dans le modèle. Les contraintes de parallélisme et d'orthogonalité commencent à influencer les objets reconstruits dans les itérations précédentes (f) La première itération avec tous les points reconstruits; (g) Itération 10: le modèle; (h) Le modèle texturé.

permet d'utiliser les méthodes standards d'algèbre linéaire pour analyser les systèmes d'équations engendrés par les contraintes. Ces systèmes sont majoritairement sous- ou sur-contraints (ou les deux au même temps, en fonction des sous-ensembles de variables). Nous avons proposé des algorithmes pour détecter les sous-ensembles sous-contraints, ce qui permet de déterminer certaines variables même dans un système sous-contraint. Nous avons aussi proposé une approche pour gérer les systèmes sur-contraints. Des exemples sur des images réelles confirment l'intérêt de ces algorithmes.

References

- [1] A. Bjorck. *Numerical Methods For Least Squares Problems*, SIAM, 1990.
- [2] S. Carlsson. *The Double Algebra: An Effective Tool for Computing Invariants in Computer Vision*, Applications of Invariance in Computer Vision: Joint European-US Workshop, pp. 145-164, 1993.
- [3] R. Cipolla and E. Boyer. *3D Model Acquisition from Uncalibrated Images*, IAPR Workshop on Machine Vision Applications, pp. 559-568, 1998.
- [4] A. Criminisi, I. Reid and A. Zisserman. *Single View Metrology*, ICCV, pp. 434-442, 1999.
- [5] P.E. Debevec, C.J. Taylor and J. Malik. *Modeling and Rendering Architecture from Photographs: a Hybrid Geometry-and Image-Based Approach*, SIGGRAPH, pp. 11-20, 1996.
- [6] O. Faugeras, Q. Luong and T. Papadopoulos. *The Geometry of Multiple Images*, MIT Press, 2001.
- [7] P.E. Gill, W. Murray and M.H. Wright. *Practical Optimization*, Academic Press, 1981.
- [8] G. Golub and C. van Loan. *Matrix computations*, The Johns Hopkins University Press Ltd., 1996.
- [9] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.
- [10] O. Henricsson. *Analysis of Image Structures Using Color Attributes and Similarity Relations*, PhD Thesis, ETH Zürich, May 1996.

- [11] S. Heuel. *Points, Lines and Planes and their Optimal Estimation*, DAGM, 2001.
- [12] P. Poulin, M. Ouimet and M.-C. Frasson. *Interactively Modeling with Photogrammetry*, Eurographics Workshop on Rendering, pp. 93-104, 1998.
- [13] H. Press, S. Teukolsky, W. Vetterling and B. Flannery. *Numerical Recipes in C*, Cambridge University Press, 1992.
- [14] D. Bondyfalat, T. Papadopoulo and B. Mourrain. *Using Scene Constraints during the Calibration Procedure*, ICCV, pp. 124-130, 2001.
- [15] P. Sturm. *Algorithms for Plane-Based Pose Estimation*, CVPR, pp. 1010-1017, 1997.
- [16] P.F. Sturm and S.J. Maybank. *A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images*, BMVC, pp. 265-274, 1999.
- [17] P. Sturm and S. Maybank. *On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications*, CVPR, pp. 432-437, 1999.
- [18] M. Wilczkowiak, E. Boyer and P. Sturm. *Camera Calibration and 3D Reconstruction from Single Images Using Parallelepipeds*, ICCV, pp. 142-148, 2001.
- [19] D. Liebowitz and A. Zisserman, *Combining Scene and Auto-calibration Constraints*, ICCV, pp.293-300, 1999.

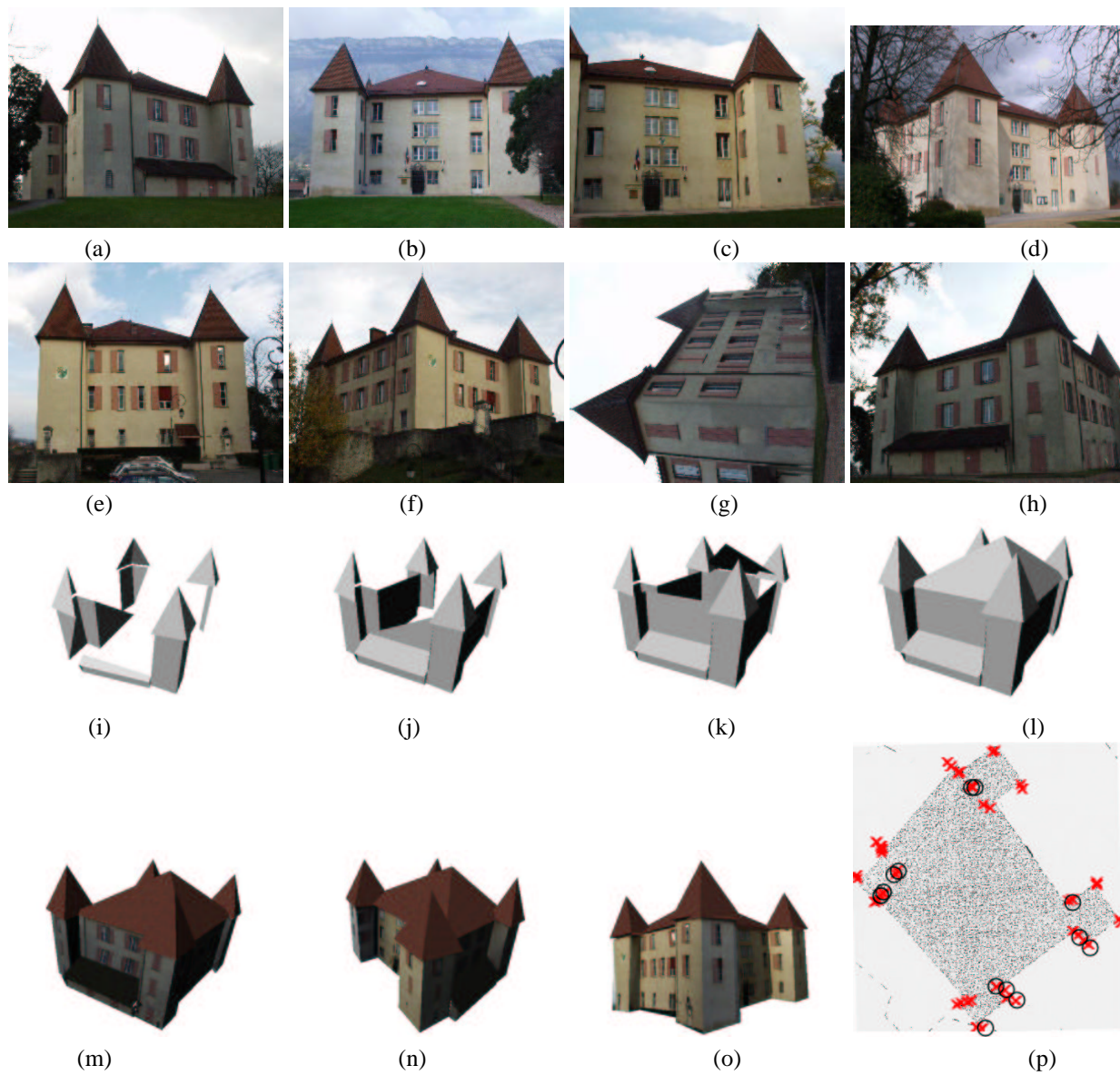


Figure 4: (a)–(h) Les photos utilisées pour la modélisation ; (i)–(l) Le parties du modèle reconstruites dans les quatre premières itérations ; (m)–(o) Les images du modèle texturé ; (p) Le plan du château avec les reprojections des points constituant les murs du château. Les points dans les cercles ont été reconstruits à l’aide de contraintes géométriques.