Université Blaise Pascal - Clermont-Ferrand II

École Doctorale
Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse présentée par :
Hanna Martinsson

Formation Doctorale CSTI :
Composants et Systèmes pour le Traitement de l'Information

en vue de l'obtention du grade de

Docteur d'Université

spécialité : Vision par ordinateur

Reconstruction d'objets manufacturés à partir de séquences d'images

Reconstructing Manufactured Objects from Image Sequences

Soutenue publiquement le 11 septembre 2008 devant le jury :

M. Jan Olof Eklundh	Président
M. Ernest Hirsch	Rapporteur
M. Nikos Paragios	Rapporteur
Mme Lourdes Agapito	Examinateur
M. François Gaspard	Co-encadrant
M. Adrien Bartoli	Co-encadrant
M. Jean-Marc Lavest	Directeur de thèse

Abstract

Three-dimensional reconstruction of objects based on a set of images is a well-known issue in the area of computer vision. It is dubbed Structure-from-Motion. In this thesis, we present an application dedicated to dimensional control in an industrial context. The constraints related to this field are strong and the requirements in terms of precision are high. Existing methods in the field of optical metrology do not allow the precise reconstruction needed, unless using tools such as the projection of structured light or the application of coded markers on the object. The idea is to reconstruct an industrial object in 3D and to compare the reconstruction to the CAD model in order to identify potential anomalies. Whereas the reconstructed surface elements describe the general aspect of the object, the discontinuities, due to holes, edges or chamfers, are not correctly depicted. To complete the reconstruction, we also consider the parametric curves extracted from the CAD model. The reconstruction will be performed in two stages, based on surface elements and on parametric curves respectively.

We present two methods dedicated to the precise reconstruction of industrial objects. The first one constitutes a brick in the process of pointwise reconstruction using an affine camera model, intended to initialize more precise and thus computationally expensive methods. A second method is dedicated to the reconstruction of parametric curves based on the observed edges in the images.

The results of the experimental evaluation of the reconstruction algorithms, using synthetic data as well as real images, are encouraging. Tests in an industrial context show that a reconstruction to the required precision can be obtained and that anomalies get identified.

Résumé

La reconstruction 3D d'objets basée sur des séquences d'images est un sujet important dans le domaine de la vision par ordinateur. Il est aussi connu sous le nom de Structure-from-Motion, structure à partir de mouvement. Dans cette thèse, nous présentons une application dédiée au contrôle de conformité dans un contexte industriel. Les contraintes du domaine sont fortes et la précision demandée est élevée. Dans le domaine de la métrologie optique, des méthodes existantes atteignent cette précision uniquement assistées de procédés tels la projection d'une lumière structurée sur l'objet ou l'utilisation de cibles codées. L'idée est de reconstruire un objet industriel en 3D et de comparer cette reconstruction au modèle CAO, afin d'identifier des anomalies éventuelles. Tandis que la reconstruction de points de surface donne l'aspect général de l'objet, les discontinuités, telles que les trous, les contours ou les chanfreins, ne sont pas bien gérées. Afin de compléter la reconstruction, nous considérons aussi les courbes paramétriques de l'objet. La reconstruction se fera ainsi en deux étapes. Elle sera basée sur les éléments de surface et sur les courbes paramétriques.

Nous présentons deux méthodes destinées à la reconstruction précise d'objets industriels. Une première méthode s'insère dans une procédure de reconstruction par points à partir d'un modèle affine de la caméra. Cette étape vise à initialiser des méthodes plus précises, avec une complexité supérieure. Une deuxième méthode est consacrée à la reconstruction de courbes paramétriques. Elle est basée sur des observations de contours dans les images.

L'évaluation expérimentale des algorithmes de reconstruction a été effectuée sur des données de synthèse ainsi que sur des images réelles dans un milieu industriel. Les résultats sont encourageants, puisque les tests montrent que la précision requise peut être obtenue. La méthode nous permet aussi de détecter des anomalies présentes sur l'objet.

Contents

Introduction

Computer vision is the scientific discipline that deals with the extraction of information by analyzing one or several images [17, 18, 31]. Whereas the functioning of the human eye is mimicked by the camera, the emulation of the entire visual system is the scope of computer vision. The beginning of computer vision goes back to the late 1970s, even though earlier work exist. It was not until the development of computers allowed the processing of large amounts of data, such as images, that the area gained in importance. Closely related fields are machine vision, robot vision, image analysis and image processing. The overlap can sometimes be so important that only the applications allows a differentiation. The applications include qualitative tasks, such as object recognition, event detection within the framework of visual surveillance or process control, the automatic navigation of vehicles, as well as quantitative tasks, such as metrology. From our point of view, image processing techniques will be used in a preliminary step, as a resource. Several mathematical tools will also be included in the algorithms needed to interpret the observed scene.

Photogrammetry is the technique of dimensional measurement from images. The requested output from a photogrammetric process is typically a map, a drawing or a 3D reconstruction of a scene or an object. The field can be classified into two disciplines: aerial photogrammetry, using an airborne camera, usually pointed at the ground, and close range photogrammtry, where the camera is close to the subject. Whereas the first serve to create topological maps, the second is used to produce 3D models of structures, vehicles, industrial objects, etc. Although the technology is as old as photography itself, it can also be considered part of the computer vision field [37, 2, 22].

Quality Control

In the context of quality control in the manufacturing industry, the systems used at present are generally based on measures obtained by contact sensors.

Coordinate measuring machines (CMMs) collect detailed dimensional data by moving a sensing device, a probe, along the surfaces of the object. Using such a process to control the conformity of a complex object is a time-consuming task and is therefore not feasible to let all objects on an assembly line undergo a control of this kind. Only the first article (First Article Inspection, FAI) or a sample will be tested off-line. Furthermore, these machines are often very expensive and are therefore not suitable in the case of a small scale production. In some contexts, the contact with reference points at the object can also be an issue, due to accessibility problems or to the environment, being hot or otherwise hostile.

Photogrammetry

Rather than using contact sensors to obtain a precise measure of the workpiece, quality control can be performed by 3D reconstruction of the object. This reconstruction can then be compared to the CAD model. The use of optical metrology, namely the projection of structured light, has proved efficient for the reconstruction of surface elements, see [53]. The use of structured light consists of projecting a light pattern, such as a grid, at a known angle onto an object. This technique can be very useful for imaging and acquiring dimensional information. Indeed, when viewed from a different angle, the observed distortions in the pattern can be translated into height variations. Scanning the object with the light yields 3D information about the shape of the object, in the form of a dense point cloud. However, the precise localization of discontinuities, such as holes or edges, remains a problem. This is due to the fact that, although arbitrarily dense, the point cloud constitutes a sampling of the surface and needs to be interpolated in order to give a continuous representation of the surface. The interpolation causes the discontinuities to be smoothed out, which obviously will be a problem. Moreover, the use of projected light close to the discontinuities of the object is problematic, due to the brusque changes in orientation of the surface. Another drawback is the need for expensive optical equipment, such as a laser.

Multi-camera photogrammetry is based upon the idea of synchronizing two (or more) cameras mounted on a calibrated bench. Coupled with a light projection tool, the typical accuracy of such a technique is of approximately 1/30000 of the largest dimension of the object. Purely optical methods, such as a laser tracker, achieve an accuracy close to 20 μm.

Other methods, derived in the context of computer vision, based on the use of coded markers to localize a known object, are present in industrial systems as well as in research projects [50]. Nevertheless, in the manufacturing industry, adapting the environment or interfering in the manufactur-

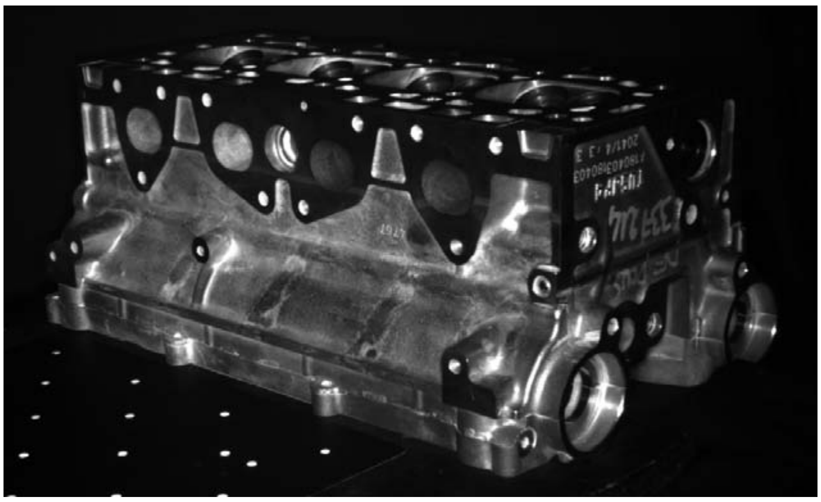


Figure 1: A cylinder head illustrates a typical manufactured object, presenting numerous details and specularities.

ing process should obviously be avoided as far as possible. Texture-based 3D reconstruction [64] has been able to reconstruct objects rather precisely. However, manufactured pieces, often metallic, are not well adapted to this kind of approach. The texture will indeed depend on the moulding and on the surroundings, due to the problem of specularities. See figure 1 for an example of an industrial object presenting specularities. A more well-suited tool is the curve-based 3D reconstruction of objects, that have been used in the computer vision community by several authors [5, 10, 55, 9, 71, 34]. A point-wise reconstruction, [55, 9], of a curve based on the triangulation of 2D estimations allows one to use results from the area of active contours [36]. Yet, in spite of a satisfactory reconstruction, this method is not optimal as it yields a curve given by an ordered point cloud, as opposed to the parametric description given by the CAD model. Others, [10, 71], have used 2D parametric curves in order to acquire a 3D curve by reprojection. The reconstruction thus obtained is clearly parametric. However, the 2D curve estimation followed by the approximation induced by the assembly of a set of curves in order to end up with a single 3D curve deteriorates the precision. A reconstruction method for parametric 3D curves of fixed complexity, using an image based criterion is presented in [5].

The above-cited works, though carried out for wide-ranging applications, allow the reconstruction of a 3D object. Nevertheless, none of them is able to reconstruct a manufactured object with the precision necessary to perform a dimensional conformity control, comparing the object to its CAD model in the case of complex 3D objects. It is in this context that our work is positioned. It consists in developing an image-based 3D reconstruction method designed for manufactured objects that is precise, robust and automatic.

Our Method

Although industrial methods for the reconstruction of surfaces are available, we are interested in finding alternative solutions, eliminating the need for expensive and bulky machinery. This can be achieved by developing suitable computer vision algorithms. Existing methods, known as bundle adjustment [65], based on a dense point-wise matching between pairs of images, proceeding by the minimization of a nonlinear cost function, need a good initialization to converge. Since the field of methods solving the nonlinear minimization problem is rather well explored in the case of 3D points, we reformulate using a bilinear factorization problem that can be solved using singular value decomposition, in order to initialize standard algorithms that will bring the adequate accuracy. The first stage of our work has therefore been to develop a bilinear method, meant to serve as a building block in a hierarchical approach to Structure-from-Motion [21]. We present a linear, thus fast, factorization based algorithm. So as to gain access to a wider range of data sets, the idea has been to extend our method to data sets with missing data points. The alignment process has therefore been embedded in an iterative structure, where estimation of the missing data and minimization based on the completed data set have been alternated. Our work on this subject has been published in [4, 41, 42] and is presented in Chapter 3.

This suboptimal reconstruction stage, intended to initialize a more precise nonlinear method, aiming at a point-wise reconstruction of the surfaces of the object, has been followed by a 3D curve reconstruction stage. The idea has been to develop a method that reconstructs the curves of the observed 3D object, based on the parametric curves extracted from the CAD model. The evolution of the curves is performed via the modification of its parameters, namely its control points, induced by the curves observed in the images. We have for this purpose defined an optimization function based on the residual 2D image error. After having developed a reconstruction method for a given number of parameters, a natural next step was to increase the number of degrees of freedom, in order to enhance the precision. This has been accomplished by successive control point insertion. The complete algorithm, including the fixed size estimation step as well as its integration in an iterative framework, has been published in [45]. So as to improve the robustness due to a failing contour detection, a different optimization function, based on an energy formulation has been considered. The reconstruction method that follows has been published in [43, 44]. We present it, together with a comparison and a conclusion of the different curve reconstruction algorithms introduced, in Chapter 4.

The final step of our work is an experimental evaluation of our method for the reconstruction of parametric curves. For the testing, we have used simulated data as well as real images. Simplified images have allowed the estimation of the theoretical limits of our algorithms, while real images of manufactured objects have served to evaluate the performance in an industrial context. The evaluation is presented in Chapter 5.

Introduction

La vision par ordinateur traite de l'extraction d'information par le biais de l'analyse d'une ou plusieurs images [17, 18, 31]. Tandis que le fonctionnement de l'œil humain est imité par la caméra, l'émulation du système visuel entier est le domaine de la vision par ordinateur. Même si des travaux antérieurs existent, les débuts de la vision par ordinateur datent de la fin des années 1970. En effet, le développement des ordinateurs a alors permis de traiter des quantités de données très grandes, telles que des images et le domaine a pris de l'importance. Dans ce vaste domaine on trouve la vision pour la robotique, le traitement et l'analyse d'images. Les recoupements entre ces activités sont tels que seulement les applications permettent de faire la différence. Ces applications peuvent être qualitatives, comme la reconnaissance d'objets, la détection d'évènements dans le cadre de la vidéosurveillance ou dans le contrôle de procédés. Les applications peuvent également être quantitatives, comme la métrologie ou la navigation automatique d'un véhicule. Les algorithmes de vision par ordinateur se basent généralement sur des techniques empruntées au traitement d'images et plusieurs outils mathématiques sont également intégrés dans les algorithmes permettant d'interpréter la scène observée.

La photogrammétrie est la technique permettant d'effectuer des mesures dimensionnelles à partir d'images et historiquement à partir de photographies. Les applications traditionnelles de la photogrammétrie sont la cartographie, le relevé de patrimoine historique ou, plus généralement, la modélisation 3D d'une scène ou d'un objet. Le domaine est généralement partagé en deux disciplines : la photogrammétrie aérienne et la photogrammétrie terrestre ("close range photogrammetry"), qui utilise une caméra proche du sujet. Tandis que le premier sert à créer des cartes topographiques, le deuxième est utilisé pour modéliser en 3D des structures, des véhicules, des objets industriels, etc. Bien que la technique date des débuts même de la photographie, elle peut aussi être considérée comme une discipline de la vision par ordinateur [37, 2, 22].

Contrôle de conformité

Dans le domaine du contrôle de conformité industriel, les systèmes utilisés actuellement sont généralement basés sur des mesures par contact. Des Machines de Mesures Tridimensionnelles (MMT ou “CMMs” en anglais) sont utilisées afin de recueillir de données dimensionnelles denses via un palpeur qui est déplacé le long des surfaces de l’objet. Le recours à ce procédé pour le contrôle de conformité d’un objet complexe nécessite beaucoup de temps et il n’est donc pas possible de contrôler l’ensemble de la production en chaine de fabrication. Seul le premier article (“First Article Inspection”, FAI) ou un échantillonnage de la production est alors mesuré hors-ligne, en plus, les machines utilisées sont souvent encombrantes et chères. Le contact avec des points de référence de l’objet peut aussi poser un problème dans certains contextes, dû à des problèmes d’accessibilité ou à un environnement chaud ou hostile.

Photogrammétrie

Les systèmes de vision offrent une alternative intéressante aux systèmes de contrôle avec des machines à mesurer par le biais de palpeurs. En effet, le contrôle de conformité peut être effectué via une reconstruction 3D de l’objet à partir de clichés pris depuis différents points de vue. Cette reconstruction, basée sur des données image, peut alors être comparée au modèle CAO ou à la gamme de mesure afin de détecter les éventuelles anomalies. L’utilisation de la métrologie optique, en particulier la projection d’une lumière structurée, s’est avérée efficace pour la reconstruction d’éléments de surface, voir [53]. Cette méthode consiste à projeter un motif, tel un quadrillage, sur l’objet. La technique peut être très efficace pour effectuer des mesures sur l’objet. En effet, les déformations du motif observées depuis des angles différents peuvent se traduire directement en variations de profondeur. Le balayage de l’objet par le faisceau génère donc l’information 3D de l’objet sous la forme d’un nuage de points dense. Cependant, la localisation précise des discontinuités, notamment les arêtes franches ou les trous et les alésages, n’est pas gérée correctement puisqu’à ces endroits la projection de lumière n’est pas possible. Ceci est dû au fait que, le nuage de points forme un échantillonnage de la surface et qu’une interpolation est nécessaire pour donner une représentation continue de la surface. L’interpolation va alors lisser les discontinuités, ce qui est bien évidemment un effet indésirable. De plus, l’utilisation de lumière projetée au niveau des discontinuités de l’objet pose problème puisqu’à ces endroits on observe des changements brusques d’orientation de la surface.

La photogrammétrie multi-caméras est basée sur l’idée de synchroniser

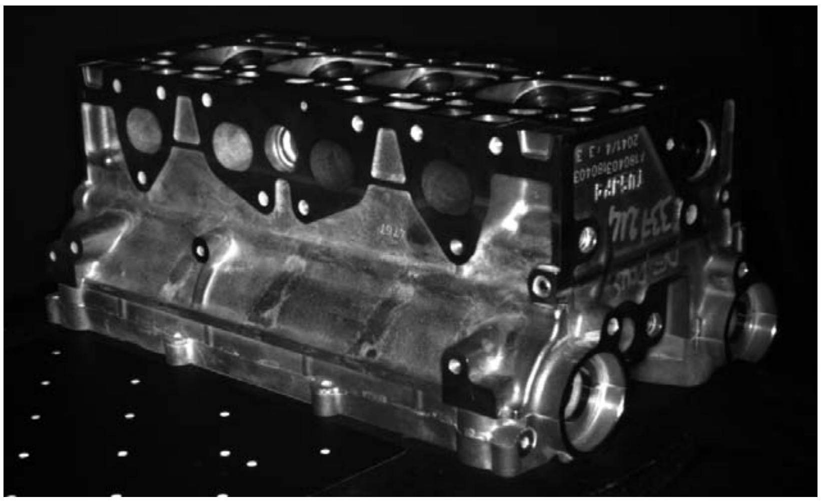


FIG. 2 – Une culasse, présentant de nombreux détails et des spécularités, sert d'exemple d'un objet manufacturé typique.

deux (ou plusieurs) caméras montées sur un banc calibré. Couplé avec un outil de projection de lumière, la précision d'une telle technique est d'environ 1/30000 de la plus grande dimension de l'objet. Des méthodes purement optiques, comme le traqueur laser, sont capables d'atteindre une précision proche de 20 μm.

D'autres méthodes, comme celles issues de la vision par ordinateur, basées sur des cibles collées sur la pièce, ont fait l'objet d'études récentes [50]. Néanmoins, les industriels souhaitent naturellement dans la mesure du possible éviter ces interventions dans le procédé de fabrication. D'autres méthodes de reconstruction 3D, basées sur la texture de l'objet [64], ont été capables de reconstruire assez précisément des objet 3D, mais les pièces manufacturées, souvent métalliques, ne sont pas adaptées à cette démarche. La texture sur ces objets métalliques dépendra en effet du procédé d'usinage (traces d'outils) et de l'environnement, dû aux problèmes de spécularités. Voir figure 2 pour un exemple d'objet industriel présentant des spécularités.

Un outil plus adapté est la reconstruction de courbes 3D, qui a été utilisé dans la communauté de la vision entre autres par [5, 10, 55, 9, 71, 34]. La reconstruction des points [55, 9] de la courbe obtenue par triangulation a également permis de tirer profit des résultats des contours actifs [36]. Or, malgré le fait que la reconstruction soit satisfaisante, cette méthode n'est pas optimale à cause d'une description de la courbe donnée par un ensemble de points, en opposition aux courbes paramétriques du modèle CAO.

D'autres [10, 71], ont utilisé des courbes paramétriques 2D afin d'aboutir à une courbe 3D par triangulation. La reconstruction ainsi obtenue est paramétrique. Cependant, l'estimation suivie d'une approximation lors de l'assemblage d'un ensemble de courbes 2D afin de former une seule courbe 3D dégrade la qualité de ces reconstructions. Une méthode de reconstruction de courbes 3D de paramétrisation donnée est présentée dans [5]. Cette méthode repose sur un critère basé image.

L'ensemble des études citées, bien qu'effectuées pour des applications variées, peut permettre de reconstruire un objet 3D. Toutefois aucune d'entre elles n'est capable de reconstruire un objet manufacturé avec la précision nécessaire pour procéder à un contrôle de sa conformité par rapport à son modèle CAO. C'est dans ce contexte que se place cette étude. Elle consiste à développer une méthode de reconstruction 3D d'objets manufacturés à partir d'images, qui soit précise, robuste et automatique.

Notre méthode

Alors que des méthodes industrielles pour la reconstruction de surfaces existent, nous cherchons des solutions alternatives qui ne reposent pas sur des machines à mesurer par contact. Le développement d'algorithmes de vision par ordinateur adaptés à ces applications nous permettra d'y parvenir. Des méthodes existantes, connues sous le nom d'ajustement de faisceaux [65], basées sur un appariement dense des point dans les différentes images, procèdent par la minimisation d'une fonction de coût non-linéaire. Ces méthodes nécessitent une initialisation proche du résultat afin de converger vers le minimum global de la fonction de cout. La résolution du problème de minimisation non-linéaire est un domaine qui a été bien exploré dans le cadre de la reconstruction de points 3D. Nous proposons donc une méthode de résolution d'un problème linéarisé, dans le but d'initialiser des algorithmes standards qui amèneront la précision requise. La première partie de nos travaux a ainsi été consacrée au développement d'une méthode linéaire, destinée à être insérée dans une approche hiérarchique au problème de Structure from Motion [21]. Nous présentons un algorithme basé sur la factorisation, méthode linéaire et donc rapide.

Une autre contribution a été d'étendre la méthode aux ensembles présentant des données manquantes. La méthode de recalage se base sur une approche itérative, où des étapes d'estimation des données manquantes et de minimisation, basée sur ces estimations, sont alternées (méthode EM). Nous avons présenté cette extension dans [4, 41, 42]. La méthode est présentée dans chapitre 2.

Cette étape de reconstruction initiale destinée à initialiser une méthode plus précise, mais non-linéaire, afin de calibrer les caméras en-ligne et de les recaler entre elles, a été suivie d'une étape de reconstruction de courbes

3D. Notre approche consiste à développer une méthode qui reconstruit les courbes de l'objet 3D à partir des courbes paramétriques du modèle CAO en faisant évoluer leurs paramètres, notamment leurs points de contrôle, en fonction des contours observés dans les images. Nous avons pour cela défini une fonction d'optimisation basée sur l'erreur résiduelle dans les images.

Après avoir développé une méthode de reconstruction pour un nombre fixe de paramètres, nous devons augmenter le nombre de degrés de liberté dans le but d'améliorer la précision de la courbe reconstruite. Ceci a été fait par insertion successive de points de contrôle. La méthode adaptative de reconstruction, basée sur la reconstruction à complexité fixée, intégrée dans un cadre itératif, a été publiée dans [45].

Afin d'être plus robuste face à une mauvaise détection de contours, une autre fonction d'optimisation, basée sur une formulation par énergie a été considérée. La méthode de reconstruction qui en découle a fait l'objet de deux publications [43, 44]. L'algorithme, ainsi qu'une comparaison et une conclusion portant sur les algorithmes différents introduits, sont présentés dans chapitre 3.

La dernière étape de nos travaux a été l'évaluation expérimentale de notre méthode de reconstruction de courbes paramétriques 3D. Pour la validation, nous avons utilisé des données de synthèse ainsi que des images réelles. Des images d'un objet simplifié nous ont permis d'estimer les limites théoriques de l'algorithme, tandis que des images réelles d'objets manufacturés complexes ont servi à évaluer les performances du système dans un contexte industriel. L'évaluation est présentée dans chapitre 4.

Chapter 1

Preliminaries

This chapter is dedicated to the presentation of various methods and algorithms that will be used in this work. Some of the techniques are classical subjects within computer vision, such as the camera model and the problem of Structure-from-Motion, whereas others, such as matrix approximation, NURBS curves and nonlinear optimization, belong to the field of applied mathematics. We will finally use a few methods from statistics.

Méthodes

Nous présentons des méthodes et algorithmes sur lesquels nos travaux vont se baser. Certaines des techniques sont classiques dans le domaine de la vision par ordinateur, tels que les modèles de caméra et le problème de reconstruction de structure à partir du mouvement, connue sous le terme de “Structure-from-Motion”, tandis que d’autres, tels que l’approximation de matrices, les courbes NURBS et l’optimisation non-linéaire, appartiennent au domaine des mathématiques appliquées. Nous allons également utiliser des méthodes statistiques et algorithmiques.

- Modèles de caméra
  Introduction aux modèles de caméra utilisés dans cette thèse.
- Approximation de matrices
  Présentation de l’approximation de faible rang d’une matrice par décomposition en valeurs singulières.
- Optimisation nonlinéaire
  Présentation de la méthode de Levenberg-Marquardt, méthode couramment utilisée pour les applications de vision par ordinateur. Cette méthode se base sur un compromis entre la rapidité de la méthode de Gauss-Newton et les propriétés de convergence de la méthode de descente de gradient.
- Structure-from-Motion
  Présentation du problématique de “Structure-from-Motion” ainsi que de quelques approches différentes.
- Courbes NURBS
  Introduction aux courbes NURBS, des B-splines non-uniformes et rationnelles. Outre une présentation des propriétés analytiques et géométriques, nous nous concentrons sur la projection perspective des courbes et sur l’insertion de points de contrôle.
- Méthodes statistiques
  Présentation de deux outils statistiques, la méthodologie d’“expectation-maximization” et le contrôle de la complexité.
- Visibilité du modèle
  Présentation d’une méthode qui permet d’identifier efficacement les parties visibles d’un objet 3D à partir d’un point de vue donné.

1.1 Notations

Vectors are typeset using bold fonts, *e.g.* $\mathbf{x}$, and matrices using capital letters, sans-serif, calligraphic and greek fonts, for example P, A, $\mathcal{Q}$ and $\mathsf{\Lambda}$. The identity matrix is denoted I and the zero matrix and vector by $\mathsf{0}$ and $\mathbf{0}$. The Frobenius or $\mathcal{L}_2$ norm of a matrix A or a vector $\mathbf{x}$ are respectively denoted $\|\mathsf{A}\|$ and $\|\mathbf{x}\|$. The Moore-Penrose pseudoinverse of matrix A is denoted $\mathsf{A}^\dagger$. Unless otherwise stated, we do not use homogeneous coordinates, that is, image point coordinates are 2-vectors: $\mathbf{x}^\mathsf{T} = (x \;\; y)$, where $^\mathsf{T}$ is transposition. A rotation matrix R is a matrix with the properties $R^{-1} = R^\mathsf{T}$ and $\det(R) = 1$. Note that in 3D space we have $R \in SO(3)$, the group of special orthogonal matrices. The mean vector of a set of vectors, say $\{\mathbf{Q}_j\}$, is denoted $\bar{\mathbf{Q}}$. Different sets of cameras are indicated with primes, *e.g.* P_1, P'_1 and P''_1 are the first cameras of the three first camera sets.

Index $i = 1 \dots n$ is used for the cameras of a camera set and index $j = 1 \dots m$ is used for the 3D points.

Calligraphic fonts are used for the measurement matrices, *e.g.*

$$\mathcal{X}_{(2n\times m)} = \begin{bmatrix} \mathcal{Y}_1 & \cdots & \mathcal{Y}_m \end{bmatrix} \qquad \text{with} \qquad \mathcal{Y}_j = \begin{bmatrix} \mathbf{x}_{1j} \\ \vdots \\ \mathbf{x}_{nj} \end{bmatrix},$$

where $\mathcal{Y}_j$ contains all the measured image coordinates for the j-th point.

1.2 Camera Models

The camera model is used in computer vision to describe the mapping of a 3D space element to the image plane. The most intuitive model is the pinhole camera, which will be presented so as to introduce the models that will be used in this thesis, the perspective and the affine camera. A more thorough survey can be found in [18, 31].

1.2.1 Pinhole Camera Model

Consider a 3D orthogonal coordinate system, with its origin in $\mathbf{C}$, which is also the optical center of the camera. The geometry of a pinhole camera is shown in figure 1.1. A 3D point $\mathbf{P} = (X, Y, Z)$ is projected onto the image plane $Z = f$ at the intersection of the line joining the point $\mathbf{P}$ with the center of projection, the optical center $\mathbf{C}$, and the image plane. The 3D coordinates of the image point are given by $(fX/Z, fY/Z, f)$. The orthogonal projection of the optical center, $\mathbf{c}$, is called the principal point and the line joining the

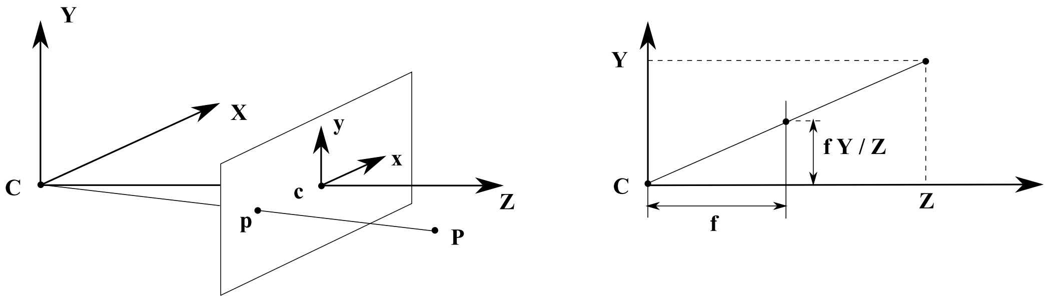


Figure 1.1: Pinhole camera geometry.

optical center and the principal point is called the principal axis. Defining a 2D coordinate system in the image plane, centered in **c**, the image coordinates of the projected point are $(fX/Z, fY/Z)$. Expressed in homogeneous coordinates, the projection is given by

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{1.1}$$

1.2.2 Perspective Camera Model

In general, points in space are expressed in a different coordinate system than that of the camera. The two systems are related by a rigid transformation, consisting of a rotation and a translation. Let R be a 3×3 rotation matrix and $\mathbf{t}$ a 3×1 translation vector. The projection matrix P is then given by

$$P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{1.2}$$

So as to take into account also internal parameters of the camera, such as the image scale (pixels per unit distance), the skew (non-orthogonal image coordinate axes) and the offset of the principal point, we consider a calibration matrix

$$K = \begin{bmatrix} \alpha_x & \tau & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1.3}$$

where $\alpha_i = fm_i$ is the focal distance in the i direction, given in pixels, (x_0, y_0) is the offset in pixels and τ is the skew. In general, τ will be zero and

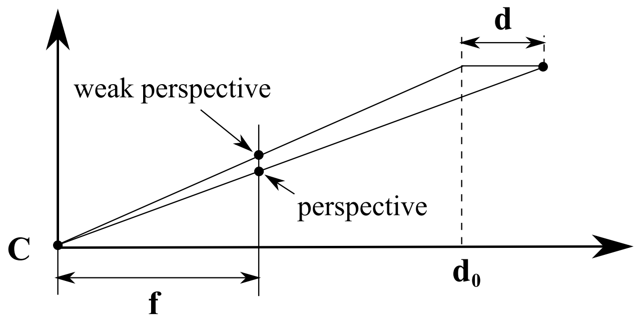


Figure 1.2: Perspective and weak perspective cameras. The projection performed by an affine camera is equivalent to an orthographic projection onto a plane $Z = d_0$, followed by a perspective projection. The difference between the affine projection and the perspective one depends on the distance from the plane, d, and on the distance from the principal axis.

$\alpha_x = \alpha_y$, which corresponds to the hypothesis of square pixels in the image sensor. The final projection matrix is then given by

$$P = \begin{bmatrix} \alpha_x & \tau & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & \mathbf{t} \end{bmatrix}. \tag{1.4}$$

1.2.3 Affine Camera Model

The mapping from the 3D point to the image plane given by the perspective camera model is widely used in computer vision. However, it presents the inconvenient of being nonlinear. In order to simplify the mathematical operations, affine cameras have been introduced. To visualize the effect of using an affine camera, we consider the technique of moving the camera center backwards along the principal axis while increasing the focal length. The result is an image with decreasing perspective effects. The approximation error due to the use of affine cameras is reasonably small if the depth of the field of view is small compared to the viewing distance. We will consider cameras with orthogonal coordinate axes, that is, having a skew equal to zero, called weak perspective cameras. An illustration of the geometry is given in figure 1.2. The effect of an affine camera compared to that of a perspective camera is shown in figure 1.3. The mathematical definition of an affine camera is that the last row of the rotation matrix is set to zero. In the

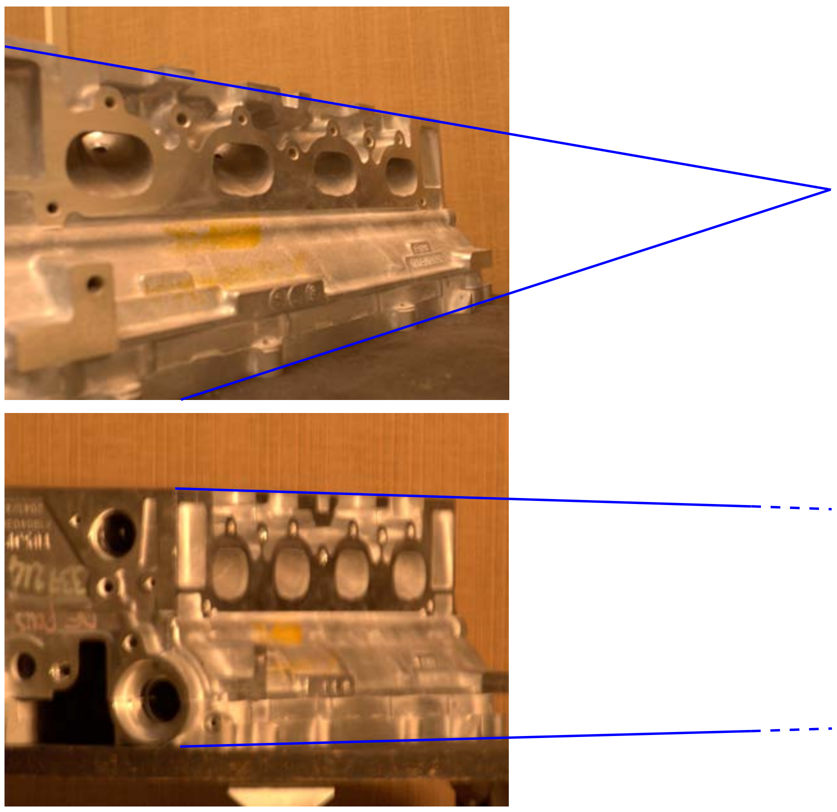


Figure 1.3: Two images of the same object with different perspective effects. The first image illustrates the effect of the perspective projection, with parallel lines joining at a point at infinity. In the second image, we have increased the focal length as well as the viewing distance. This diminishes the depth of the observed object and places the point at infinity far away.

case of a weak perspective camera, the projection matrix can be written as

$$P = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^\top & t_1 \\ \mathbf{r}_2^\top & t_2 \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{1.5}$$

Stressing the affine character of the transformation, using non-homogeneous point coordinates, the projection can be written

$$\begin{bmatrix} x \\ y \end{bmatrix} = M \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{t}, \tag{1.6}$$

where the calibration parameters are integrated in the 2×3 matrix M and the 2-vector $\mathbf{t}$.

1.3 Matrix Approximation using Singular Value Decomposition

In this section, we present a useful tool from linear algebra called the singular value decomposition. This is in order to introduce one of its applications, the low rank approximation of a matrix. For more details, please refer to [25].

1.3.1 Singular Value Decomposition

The singular value decomposition (SVD) [25] is a factorization of an arbitrary matrix. It can be seen as a generalization of the spectral theorem to rectangular matrices.

Let A be an $m \times n$ matrix. As eigenvalues are not defined for rectangular matrices, we consider the singular values, that is, the eigenvalues of the square matrix $A^{\mathsf{T}}A$. There exists a factorization of the matrix A of the form

$$A_{m\times n} = U_{m\times n}\Sigma_{n\times n}V_{n\times n}^{\mathsf{T}},$$

where U and V are orthonormal matrices and Σ is a diagonal matrix containing the singular values of A. A common convention is to order the singular values in a non-decreasing way. If this is the case, then Σ is uniquely determined by A, whereas U and V might not be.

1.3.2 Low Rank Matrix Approximation

The rank r of the matrix A is the number of non-zero singular values, that is, the diagonal elements of Σ. The r first column vectors of U form an orthonormal basis for the column space (range) of A and the r first column vectors of V form an orthonormal basis for the row space of A. Indeed, the SVD can be written as a sum of rank-one matrices

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^{\mathsf{T}}.$$

The best rank ν approximation of A, in the least squares sense, is

$$A_\nu = \sum_{i=1}^{\nu} \sigma_i u_i v_i^{\mathsf{T}},$$

with the approximation error given by $\|A - A_\nu\| = \sqrt{\sigma_{\nu+1}^2 + \cdots + \sigma_r^2}$.

1.4 Nonlinear Least Squares Optimization

A problem that often arises in numerical applications is to minimize an error expressed as a sum of squared residuals. For a set of m functions f_i, we want to solve

$$\min_x \sum_{i=0}^{m} \left(f_i(x)\right)^2 . \tag{1.7}$$

If the functions are linear, the solution can be obtained for example using singular value decomposition. However, for general nonlinear functions, iterative methods must be used.

The idea is to start with an initial (guessed) value $x^{(0)}$ and to proceed with successive updates

$$x^{(k+1)} = x^{(k)} + \delta^{(k)}. \tag{1.8}$$

The step $\delta^{(k)}$ can be computed using different methods.

1.4.1 Gradient Descent

An iterative minimization method using the first order derivative of the functions f_i is the gradient descent method. The idea is to take steps in the direction of the negative gradient, since this is the direction in which the function decreases the most rapidly. Noting $f^{(k)} = f(x^{(k)})$, the step in the k^{th} iteration is then given by

$$\delta^{(k)} = -\alpha J_k^{\mathsf{T}} f^{(k)}, \tag{1.9}$$

where α is the step length and can be set in advance. However, in more sophisticated gradient descent algorithms, the choice of α is more elaborate than that.

1.4.2 Gauss-Newton

A classical method for finding the optimum of a function is the Gauss-Newton method, which is an iterative procedure based on a linear approximation of the functions f_i. The method uses a first-order Taylor expansion of the functions f_i in a neighborhood of the current point $x^{(k)}$ to compute the step $\delta^{(k)}$. Using the evaluation of the jacobian, J_k, in the current point, $\delta^{(k)}$ is obtained by solving the linear system, called the normal equations

$$J_k^{\mathsf{T}} J_k \, \delta^{(k)} = -J_k^{\mathsf{T}} f^{(k)}. \tag{1.10}$$

We note that $\delta^{(k)}$ is the least squares solution to the linear problem

$$J_k \, \delta^{(k)} = f^{(k)}, \tag{1.11}$$

which is solved using the pseudo-inverse of the jacobian.

1.4.3 Levenberg-Marquardt

In 1944, Levenberg presented an algorithm [38], rediscovered in 1963 by Marquardt [40], that combines the two methods presented above. When far from the minimum, the gradient descent approach is favored, as it lowers the value of the cost function at all times. On the other hand, when close to the solution, the Gauss-Newton method will be privileged, due to its faster convergence. The Levenberg-Marquardt method interpolates between the two previously described methods. The linear equation system giving the step in the Gauss-Newton method is replaced by a damped version

$$\left(J_k^\mathsf{T} J_k + \lambda_k I\right) \delta^{(k)} = -J_k^\mathsf{T} f^{(k)}. \tag{1.12}$$

The parameter λ_k, called the damping parameter, is updated in each step. Several heuristics exist for the choice of λ_k.

The algorithm suggested by Marquardt consists in choosing an initial value λ_0 and a factor $\nu > 1$. In each iteration, the function is evaluated after a step using λ and another one using $\frac{\lambda}{\nu}$. If none of the two decreases the cost function value, λ is multiplied with ν until a lower value is found. If $\frac{\lambda}{\nu}$ lead to a lower function value, then this is taken a the new λ and the new optimum is the function value found using this parameter. If, on the contrary, it is the use of λ that gives the lowest function value, then it is left unchanged and the optimum is the function value associated.

In short, if the cost function has a close to linear behavior around the current point, the algorithm will favor the Gauss-Newton iteration by choosing a small λ. Otherwise, a gradient descent iteration will be used.

1.5 Structure-from-Motion

Three-dimensional reconstruction from multiple images of a rigid scene, called Structure-from-Motion, is one of the most studied problems in computer vision. The difficulties come from the fact that, using only feature correspondences, both the 3D structure of the scene and the cameras have to be computed. Most approaches rely on an initialization phase optionally followed by self-calibration and bundle adjustment [65]. More details on Structure-from-Motion can be found in [31].

1.5.1 Calibration and Geometry

The problem of reconstruction can be applied to several geometrical camera models and to calibrated as well as uncalibrated cameras. A camera is considered calibrated if its internal parameters, such as the focal length, the principal point and the distortion, are known.

Early work concerned calibrated cameras and Euclidean structure. The area has now reached the level of maturity where working systems are integrated in numerous applications, for example in the navigation of vehicles [3, 29, 28, 74, 51]. Working with calibrated cameras present the obvious advantage of being able to give the Euclidean structure of the scene. However, there are a number of drawbacks that have motivated the development of methods using uncalibrated cameras. First, camera calibration may not be available or it may be of poor quality. Camera calibration errors may induce structure errors that can be difficult to cancel out, since a nonlinear model must be considered to remove the distortion due to the camera lens. Second, the use of uncalibrated cameras allows for the camera parameters to be altered over time, for example, the focal length can be changed by zooming or focusing.

Relaxing the calibration constraint, the reconstruction obtained is projective, that is, given up to a projective transformation. Projective transformations do not preserve size and angle but do preserve incidence and cross-ratio. The nature of the projective transformation diminishes the connections between the recovered 3D structure and the physical geometry of the scene. Projective reconstruction has been described in [30, 49, 60].

A good compromise between the difficulty of computation and the information content is to use affine geometry. Affine structure is a specialization to projective structure, that can be specialized further to Euclidean structure. An affine reconstruction yields affine cameras and is defined up to an affinity. Affine invariants include planarity, parallelism and length ratios along parallel directions.

1.5.2 Sequential and Hierarchical Algorithms

Existing initialization algorithms can be divided into two families, namely sequential and hierarchical processes. Hierarchical processes [21], based on batch processes, have proven the most successful for large image sequences. Indeed, batch processes such as the factorization algorithms [59, 62] which reconstruct all features and cameras in a single computation step, need all points to be visible at all times, and are therefore more suitable as building blocks in hierarchical schemes than as independent methods. Sequential processes such as [6] which reconstruct each view on turn, may typically suffer from accumulation of the errors. In order to prevent drift over time of the coordinate frame, the authors use periodic updates based on an existing arbitrary projective coordinate frame. Hierarchical processes merge partial 3D models obtained from sub-sequences, which allows one to distribute the error over the sequence, and efficiently handle open and closed sequences. A key step of hierarchical processes is the fusion or the alignment of partial 3D models, which is done by computing 3D motion from 3D feature correspondences. This problem has been extensively studied in the projective [12, 21] and the metric and Euclidean [32, 70] cases.

1.5.3 Projective Reconstruction – Bundle Adjustment

Consider a set of images together with a set of observed point matches. Let $\mathbf{x}_j^i$ be the homogeneous coordinates of the j^{th} in the i^{th} view. We want to retrieve the projective cameras P^i and the 3D points $\mathbf{X}^j$ such that the reprojection error is zero. In the presence of noise, the equality will not be satisfied. Assuming i.i.d. Gaussian noise on the image coordinates, the Maximum Likelihood solution is obtained by estimating projection matrices P^i and 3D points $\mathbf{X}_j$, verifying $\hat{\mathbf{x}}_j^i = P^i\mathbf{X}_j$, such that the euclidean distance $d(\mathbf{x}_j^i, \hat{\mathbf{x}}_j^i)$ is minimized

$$\min_{P^i,\mathbf{X}_j} \sum_{i,j} d(\mathbf{x}_j^i, \hat{\mathbf{x}}_j^i) = \min_{P^i,\mathbf{X}_j} \sum_{i,j} d(\mathbf{x}_j^i, P^i\mathbf{X}_j). \tag{1.13}$$

Note that we use homogeneous coordinates to express $\hat{\mathbf{x}}_j^i$. This estimation, using the minimization of the 2D reprojection error is called bundle adjustment [65]. The name suggests that the method adjusts the bundle of rays passing through each camera center and the set of 3D points or, equivalently, between each 3D point and the set of camera centers.

The algorithm handles missing data and allows individual covariances for the measurements. It can further be extended to take into account priors on camera parameters and point positions. Bundle adjustment is the standard method optimizing an existing 3D model. The method presents however two major drawbacks: First, it requires a good initialization so as to avoid getting stuck in local minima. Second, the computational complexity is high, since a very large number of parameters must be estimated. The solution to the first problem is to use a faster method to compute an approximate solution to use as initialization. Affine factorization, which will be described next is one such method. Concerning the computational complexity of bundle adjustment, there are a number of different approaches that aim to reduce the complexity of the problem [21, 51].

1.5.4 Affine Reconstruction by Factorization

The factorization algorithm was introduced by Tomasi and Kanade [62]. Assuming independent identically distributed Gaussian noise on the given image coordinates, it yields a Maximum Likelihood affine reconstruction. The prerequisites are a set of images together with a set of point matches $\mathbf{x}_j^i$, visible in all views. The aim is to estimate affine cameras $\{M^i, \mathbf{t}^i\}$ and 3D points $\mathbf{X}^j$ that minimize the error in the images, that is, such that the distance between the observed points and the estimated projected points $\hat{\mathbf{x}}_j^i = M^i\mathbf{X}_j + \mathbf{t}^i$ is minimized

$$\min_{M^i,\mathbf{t}^i,\mathbf{X}_j} \sum_{i,j} \left\|\mathbf{x}_j^i - \hat{\mathbf{x}}_j^i\right\|^2 = \min_{M^i,\mathbf{t}^i,\mathbf{X}_j} \sum_{i,j} \left\|\mathbf{x}_j^i - (M^i\mathbf{X}_j + \mathbf{t}^i)\right\|^2 . \tag{1.14}$$

Since the affine transformation maps centroids to centroids, the translational part can be estimated and eliminated and we can thus take $\mathbf{t}^i = \mathbf{0}$. This can be seen as considering coordinate systems centered in the centroids, in 3D space as well as in the images. We now want to solve the minimization problem

$$\min_{M^i,\mathbf{X}_j} \sum_{i,j} \left\|\mathbf{x}_j^i - \hat{\mathbf{x}}_j^i\right\|^2 = \min_{M^i,\mathbf{X}_j} \sum_{i,j} \left\|\mathbf{x}_j^i - M^i\mathbf{X}_j\right\|^2 , \tag{1.15}$$

that can be formulated using matrices. We define the $2m \times n$ measurement matrix $\mathcal{X}$ regrouping the measured image points expressed in centered coordinates

$$\mathcal{X} = \begin{bmatrix} \mathbf{x}_1^1 & \mathbf{x}_2^1 & \cdots & \mathbf{x}_n^1 \\ \mathbf{x}_1^2 & \mathbf{x}_2^2 & \cdots & \mathbf{x}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1^m & \mathbf{x}_1^1 & \cdots & \mathbf{x}_n^m \end{bmatrix} . \tag{1.16}$$

Ideally, we would want to find matrices M^i and points $\mathbf{X}_j$ such that

$$\mathcal{X} = \begin{bmatrix} M^1 \\ M^2 \\ \vdots \\ M^m \end{bmatrix} \begin{bmatrix} \mathbf{X}^1 & \mathbf{X}^2 & \cdots & \mathbf{X}^n \end{bmatrix} . \tag{1.17}$$

Due to the structure of the right-hand side, a $2m \times 3$ motion matrix multiplied with a $3 \times n$ structure matrix, the rank is (at most) 3. In the presence of noise, the rank of $\mathcal{X}$ will in general be higher than 3 and the system will not be satisfied exactly. The reconstruction of the scene is given by a decomposition of the measurement matrix into two matrices of the given sizes, that will thus fit the rank 3 criterion. Defining similarly $\hat{\mathcal{X}} = (\hat{\mathbf{x}}_j^i)_{i,j}$ and using the Frobenius norm, we have

$$\|\mathcal{X} - \hat{\mathcal{X}}\|^2 = \sum_{i,j} \left\|\mathbf{x}_j^i - \hat{\mathbf{x}}_j^i\right\|^2 = \sum_{i,j} \left\|\mathbf{x}_j^i - M^i\mathbf{X}_j\right\|^2 . \tag{1.18}$$

Minimizing the 2D reprojection error is therefore equivalent to finding the best rank 3 approximation of $\mathcal{X}$. This is accomplished by using the SVD decomposition, that provides the approximation as well as the decomposition into a motion matrix $\hat{M}$ and a structure matrix $\hat{X}$. The decomposition is however not unique. If $\hat{\mathcal{X}} = U_{2m\times 3}\Sigma_{3\times 3}{V_{n\times 3}}^\mathsf{T}$, then we can take for example

$$\left\{ \begin{array}{rcl} \hat{M} & = & U_{2m\times 3}\Sigma_{3\times 3} \\ \hat{X} & = & {V_{n\times 3}}^\mathsf{T} \end{array} \right. \tag{1.19}$$

or

$$\left\{ \begin{array}{rcl} \hat{M} & = & U_{2m\times 3} \\ \hat{X} & = & \Sigma_{3\times 3}{V_{n\times 3}}^\mathsf{T} \end{array} \right. . \tag{1.20}$$

We also note that the insertion of an arbitrary non-singular 3×3 matrix together with its inverse does not affect the result, since

$$\hat{\mathcal{X}} = \hat{M}AA^{-1}\hat{X} = (\hat{M}A)(A^{-1}\hat{X}). \tag{1.21}$$

The reconstruction is hence given up to multiplication with a common matrix, which makes the reconstruction affine. In its original form, the factorization algorithm do not handle occlusions since complete matrices are needed. However, recent extensions have been proposed that handles the presence of missing data [61].

1.6 NURBS Curves

In computer vision, curves are generally represented either by (ordered) point clouds or by parametric functions. So as to be able to work with a limited number of parameters, we will opt for the latter. In order to satisfy multiple constraints while keeping the number of degrees of freedom low, a natural choice is to use piece-wise polynomials or piece-wise rational curves. B-spline curves are a common choice, due to their many interesting properties and the existence of numerically stable computational algorithms. B-splines are expressed as linear combinations of polynomial basis functions. NURBS (Non-Uniform Rational B-Splines) is a generalization of B-splines that presents the important property of being closed under affine and perspective transformations.

For more details on NURBS curves and their properties, refer to [52].

1.6.1 Definition

Let us start by introducing B-splines. Let $U = \{u_0, \cdots, u_m\}$ be a nondecreasing vector, called the knot vector. The B-spline basis functions $N_{i,k}(t)$ are given by the recursive relation

$$N_{i,0}(t) = \begin{cases} 1 & \text{if } u_i \leq t \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{1.22}$$

$$N_{i,k}(t) = \frac{t - u_i}{u_{i+k} - u_i} N_{i,k-1(t)} + \frac{u_{i+k+1} - t}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(t), \tag{1.23}$$

where k is the degree. A B-spline curve is now defined as a linear combination of the basis functions and the control points $\mathbf{P}_i$ by

$$\mathbf{C}(t) = \sum_{i=0}^{n} \mathbf{P}_i \, N_{i,k}(t). \tag{1.24}$$

A NURBS curve is a vector valued, piecewise rational polynomial, defined by

$$\mathbf{C}(t) = \frac{\sum_{i=0}^{n} w_i \, \mathbf{P}_i \, N_{i,k}(t)}{\sum_{i=0}^{n} w_i, N_{i,k}(t)}, \tag{1.25}$$

where w_i are weights associated with the control points. A geometrical interpretation of the weights will be given in section 1.6.3. If we define the rational basis functions

$$R_{i,k}(t) = \frac{w_i N_{i,k}(t)}{\sum_{j=0}^{n} w_j N_{j,k}(t)}, \tag{1.26}$$

the curve is written

$$\mathbf{C}(t) = \sum_{i=0}^{n} \mathbf{P}_i R_{i,k}(t). \tag{1.27}$$

The name NURBS (Non-Uniform Rational B-Splines) indicates that the knot vector is non-uniform, that is, the knot points are not equidistant, and that the pieces of the curve are rational polynomials.

1.6.2 Analytic and Geometric Properties

The analytic properties of the rational basis functions determine the geometric properties of NURBS curves. The most important properties are

- **Generalization:** If all weights are equal to 1, then $R_{i,k}(t) = N_{i,k}(t)$.
- **Locality:** $R_{i,k}(t) = 0$ if $t \notin [u_i, u_{i+k+1})$.

- **Partition of unity:** $\sum_i R_{i,k}(t) = 1$
- **Differentiability:** Inside a knot span, the basis function is a rational polynomial, so it is infinitely continuously differentiable if the denominator is bounded away from zero. At knots, it is $k-p$ times continuously differentiable, where p is the multiplicity of the knot.

The following geometric properties of NURBS curves arise from the aforementioned properties of the basis functions

- B-spline curves is a special case of NURBS curves.
- **Local approximation:** If a control point is moved or a weight is changed, this will affect the curve only on $k + 1$ intervals.
- **Strong convex hull:** If $t \in [u_i, u_{i+1})$, then $\mathbf{C}(t)$ is in the convex hull of the control points $\mathbf{P}_{i-k}, \ldots, \mathbf{P}_i$, supposing nonnegative weights.
- **Covariance:** Covariance under affine and perspective transformations.
- **Differentiability:** NURBS curves have the same differentiability properties as the rational basis functions.
- **Variation diminishing:** No hyperplane intersects the NURBS curve more times than it intersects the polygon formed by its control points, supposing nonnegative weights.
- If $m+1$ is the number of knots, $n+1$ the number of control points and k the degree, then the equality $m = n + k + 1$ holds.

It is a common choice to take $k = 3$, which has proved to be a good compromise between required smoothness and the problem of oscillation, inherent to high degree polynomials. Note that the degree of a NURBS curve defines the number of control points that influence any given point on the curve. Since the region of influence of a control point is delimited by the knot points, we can talk in terms of intervals. For any given interval, there are $k + 2$ control points that have an impact on the associated curve section. Inversely, for any given control point, there are $k+2$ intervals where it impacts the curve.

For the parameterization of closed curves, we consider periodic knot vectors, that is, verifying $u_{j+p} = u_j$ for some p. Following the reasoning outlined above, for a closed curve there must be as many control points as there are intervals of non-zero length. If we take into account the equality $m = n+k+1$, we see that the knot vector contains more knots than that. The reason to this is the overlapping intervals, needed to close the curve in a smooth way.

Given all these parameters, the set of NURBS defined on U forms, together with the operations of point-wise addition and multiplication with a scalar, a vector space.

1.6.3 Geometrical Interpretation of NURBS Curves

In order to provide a geometrical interpretation of the weights associated with the control points and of the curve itself, we take the example of a B-spline curve in space and its corresponding NURBS curve in a given plane. Consider the Euclidean 3D space, with coordinate axes X, Y and W, and the projective plane, with coordinate axes x and y, as pictured in figure 1.4. Any point in the X, Y, W coordinate system can be written as

$$\mathbf{P}^w = \begin{cases} (xw, yw, w) & \text{if} \quad w \neq 0 \\ (x, y, 0) & \text{if} \quad w = 0 \end{cases} \tag{1.28}$$

and can be mapped onto the plane by the perspective transformation

$$T(\mathbf{P}^w) = \begin{cases} (x, y) & \text{if} \quad w \neq 0 \\ \text{direction}\,(x, y) & \text{if} \quad w = 0. \end{cases} \tag{1.29}$$

For a set of 3D control points $\mathbf{P}_i^w = (x_i w_i, y_i w_i, w_i)$ and a given knot vector, we can construct the non-rational B-spline curve in 3D space

$$\mathbf{C}^w(t) = \sum_{i=0}^{n} \mathbf{P}_i N_{i,k}(t). \tag{1.30}$$

Writing out the coordinate functions, we get

$$\begin{cases} X(t) &= \sum_{i=0}^{n} w_i x_i \, N_{i,k}(t) \\ Y(t) &= \sum_{i=0}^{n} w_i y_i \, N_{i,k}(t) \\ Z(t) &= \sum_{i=0}^{n} w_i \, N_{i,k}(t) \end{cases} . \tag{1.31}$$

The curve can be mapped onto the plane, see figure 1.4, giving the projected curve, expressed coordinate-wise

$$\begin{cases} x(t) &= \dfrac{X(t)}{Z(t)} = \dfrac{\sum_{i=0}^{n} w_i x_i \, N_{i,k}(t)}{\sum_{i=0}^{n} w_i \, N_{i,k}(t)} \\ y(t) &= \dfrac{Y(t)}{Z(t)} = \dfrac{\sum_{i=0}^{n} w_i y_i \, N_{i,k}(t)}{\sum_{i=0}^{n} w_i \, N_{i,k}(t)} \end{cases} \tag{1.32}$$

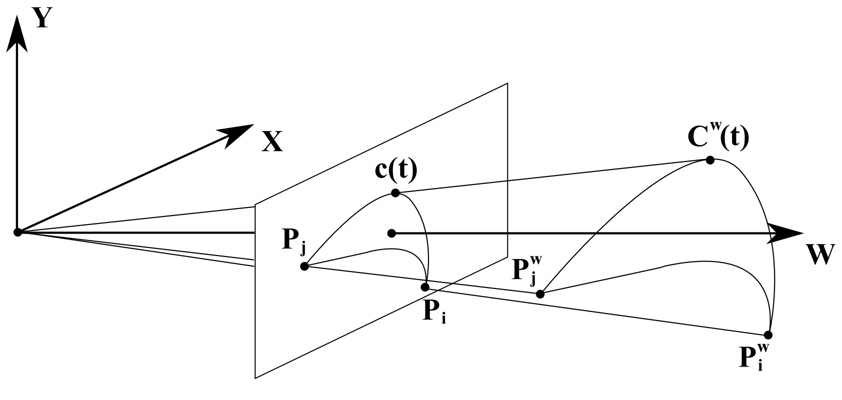


Figure 1.4: Geometric construction of 2D NURBS curves based on a perspective projection of a 3D B-spline curve.

Using vector notation, we finally obtain

$$\mathbf{c}(t) = T\left(\mathbf{C}^w(t)\right) = \frac{\sum_{i=0}^{n} w_i\, T(\mathbf{P}_i^w)\, N_{i,k}(t)}{\sum_{i=0}^{n} w_i,\, N_{i,k}(t)} = \frac{\sum_{i=0}^{n} w_i\, \mathbf{P}_i\, N_{i,k}(t)}{\sum_{i=0}^{n} w_i,\, N_{i,k}(t)}. \tag{1.33}$$

We note that the perspective transformation of a 3D B-spline yields a 2D NURBS curve. Indeed, using homogeneous notation, the 3D B-spline curve is an alternative way of expressing the 2D NURBS curve, the added coordinate serving as weight. In the following section, we will see that the perspective transformation maps NURBS to NURBS.

1.6.4 Perspective Projection of NURBS Curves

According to the perspective camera model, see Sec. 1.2, the projection $T(\cdot)$ that transforms a world point into an image point is expressed using homogeneous coordinates by means of the transformation matrix $T_{3\times 4}$ as

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = T_{3\times 4} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} T_{1,1}X + T_{1,2}Y + T_{1,3}Z + T_{1,4} \\ T_{2,1}X + T_{2,2}Y + T_{2,3}Z + T_{2,4} \\ T_{3,1}X + T_{3,2}Y + T_{3,3}Z + T_{3,4} \end{bmatrix}. \tag{1.34}$$

In (non-homogeneous) image coordinates, we have

$$\begin{cases} x = \dfrac{\tilde{x}}{\tilde{z}} = \dfrac{T_{1,1}X + T_{1,2}Y + T_{1,3}Z + T_{1,4}}{T_{3,1}X + T_{3,2}Y + T_{3,3}Z + T_{3,4}} \\ y = \dfrac{\tilde{y}}{\tilde{z}} = \dfrac{T_{2,1}X + T_{2,2}Y + T_{2,3}Z + T_{2,4}}{T_{3,1}X + T_{3,2}Y + T_{3,3}Z + T_{3,4}} \end{cases}. \tag{1.35}$$

Mapping the curve to the image plane yields, in the x coordinate

$$x(t) = \frac{\tilde{x}(t)}{\tilde{z}(t)} = \frac{\sum_i w_i (T_{1,1}X_i + T_{1,2}Y_i + T_{1,3}Z_i + T_{1,4}) R_{i,k}(t)}{\sum_i w_i (T_{3,1}X_i + T_{3,2}Y_i + T_{3,3}Z_i + T_{3,4}) R_{i,k}(t)}. \tag{1.36}$$

Defining the new weights

$$w'_i = (T_{3,1}X_i + T_{3,2}Y_i + T_{3,3}Z_i + T_{3,4})\, w_i \tag{1.37}$$

and recalling the expression for the rational basis functions given in 1.26, we have

$$\begin{aligned} x(t) &= \frac{\sum_i w'_i \frac{T_{1,1}X_i + T_{1,2}Y_i + T_{1,3}Z_i + T_{1,4}}{T_{3,1}X_i + T_{3,2}Y_i + T_{3,3}Z_i + T_{3,4}} \frac{N_{i,k}(t)}{\sum_j w_j N_{j,k}(t)}}{\sum_i w'_i \frac{N_{i,k}(t)}{\sum_j w_j N_{j,k}(t)}} \\ &= \frac{\sum_i w'_i \, x_i \, N_{i,k}(t)}{\sum_i w'_i \, N_{i,k}(t)} \end{aligned} . \tag{1.38}$$

The expression for $y(t)$ is derived similarly. Using vector notation, the projected curve is written

$$\mathbf{c}(t) = T(\mathbf{C})(t) = \frac{\sum_{i=0}^{n} w'_i \, T(\mathbf{P}_i)\, N_{i,k}(t)}{\sum_{i=0}^{n} w'_i \, N_{i,k}(t)} = \sum_{i=0}^{n} T(\mathbf{P}_i) R'_{i,k}(t), \tag{1.39}$$

where the $R'_{i,k}$ are the basis functions of the projected NURBS.

Deriving the formula for the new weights using a geometrical formulation yields the expression

$$w'_i = \mathbf{n} \cdot (\mathbf{P}_i - \mathbf{C}_{\mathrm{O}})\, w_i, \tag{1.40}$$

where $\mathbf{n}$ is a unit vector along the optical axis and $\mathbf{C}_{\mathrm{O}}$ the optical center of the camera.

1.6.5 Control Point Insertion

One of the fundamental geometric algorithms available for NURBS curves is the control point insertion. The key is the knot insertion, which is equivalent to adding one dimension to the vector space, consequently adapting the basis. Since the original vector space is included in the new one, there is a set of control points such that the curve remains unchanged.

Let $\bar{u} \in [u_j, u_{j+1})$. We insert $\bar{u}$ in U, forming the new knot vector

$$\bar{U} = \{\bar{u}_0 = u_0, \cdots, \bar{u}_j = u_j, \bar{u}_{j+1} = \bar{u}, \bar{u}_{j+2} = u_{j+1}, \cdots, \bar{u}_{m+1} = u_m\}.$$

The new control points $\bar{\mathbf{P}}_i$ are given by the linear system

$$\sum_{i=0}^{n} \mathbf{P}_i R_{i,k}(t) = \sum_{i=0}^{n+1} \bar{\mathbf{P}}_i \bar{R}_{i,k}(t). \tag{1.41}$$

Due to the local influence of the basis functions and to the fact that $\bar{u} \in [u_j, u_{j+1})$, we have

$$\sum_{i=j-k}^{j} \mathbf{P}_i R_{i,k}(t) = \sum_{i=j-k}^{j+1} \bar{\mathbf{P}}_i \bar{R}_{i,k}(t). \tag{1.42}$$

for $u \in [u_j, u_{j+1})$ and

$$\begin{cases} R_{i,k}(t) = \bar{R}_{i,k}(t) & 0 \leq i \leq j-k-1 \\ R_{i,k}(t) = \bar{R}_{i+1,k}(t) & j+1 \leq i \leq n \end{cases} \tag{1.43}$$

Equations (1.42) and (1.43) together with the linear independence of the basis functions imply that

$$\begin{cases} \mathbf{P}_i = \bar{\mathbf{P}}_i & 0 \leq i \leq j-k \\ \mathbf{P}_i = \bar{\mathbf{P}}_{i+1} & j \leq i \leq n \end{cases} \tag{1.44}$$

It can be shown, by induction on k, that for $j-k \leq i \leq j+1$

$$R_{i,k}(t) = \frac{\bar{u} - \bar{u}_i}{\bar{u}_{i+p+1} - \bar{u}_i} \bar{R}_{i,k}(t) + \frac{\bar{u}_{i+p+2} - \bar{u}}{\bar{u}_{i+p+2} - \bar{u}_{i+1}} \bar{R}_{i+1,k}(t). \tag{1.45}$$

Substituting (1.42) and (1.45), using the knot vector U instead of $\bar{U}$, yields after some calculations

$$\bar{\mathbf{P}}_i = \alpha_i \, \mathbf{P}_i + (1 - \alpha_i) \, \mathbf{P}_{i-1}, \tag{1.46}$$

with

$$\alpha_i = \begin{cases} 1 & & i \leq j-k \\ \dfrac{\bar{u} - u_i}{u_{i+k} - u_i} & \text{if} & j-k+1 \leq i \leq j \\ 0 & & i \geq j+1 \end{cases} . \tag{1.47}$$

Note that only k new control points need to be computed, due to the local influence of splines.

1.7 Statistical Methods

Although our algorithms are completely deterministic, there are a few statistical methods that will be useful. The first one, the Expectation-Maximization algorithm, is used for parameter estimation in probabilistic models, whereas the second one is used to determine the appropriate model complexity in a model estimation framework.

1.7.1 Expectation-Maximization

The Expectation-Maximization (EM) algorithm was presented and named in 1977 in an article by Dempster et al. [14]. Although the methodology had already been used in different settings, Dempster et al. generalized it and developed the underlying theory. Indeed, EM is a description of a class of related algorithms, rather than a specific algorithm.

An EM algorithm is an iterative statistical method for finding Maximum Likelihood (ML) estimates of parameters in probabilistic models, where the model depends on unobserved data, that is, given an incomplete set of measurement data. The main idea is to alternate between predicting the missing data and estimating the model. When the log likelihood cannot be maximized, for example due to missing data, it is replaced by its conditional expectation given the observed data, using the current estimate of the parameters. The conditional expectation at iteration k is written

$$E^{(k)}(\Theta) = E\left[\,\log L(\Theta) \,|\, \mathcal{X}_{\text{obs}}, \Theta^{(k)}\,\right], \tag{1.48}$$

where Θ is the set of parameters, $\log L(\Theta)$ the log likelihood and $\mathcal{X}_{\text{obs}}$ the observed data. In the maximization step, this conditional expectation is then maximized with respect to the parameters and the new parameter vector is given by

$$\Theta^{(k+1)} = \arg\max_{\Theta} E^{(k)}(\Theta). \tag{1.49}$$

In the case where the log likelihood is a linear function of the missing data, this simply consists in replacing the missing data by their conditional expectations given the observed data at current parameter values. This approximate log likelihood is then maximized so as to yield a new estimate of the parameters. The log likelihood increases in each iteration and the process converges to a local minimum of the approximate Maximum Likelihood residual error. The rate of convergence of the EM algorithm is linear and a function of the rate of missing data. See *e.g.* [46] for details and proof of convergence.

An EM algorithm can also be used to find a maximum a priori (MAP) estimate, by performing MAP estimation in the maximization step instead of maximum likelihood.

1.7.2 Handling Complexity for Parametric Models

In general, an optimization process is terminated when convergence of some error function is obtained. When dealing with model estimation, a common measure of the quality of a particular model is the residual, that is, the difference between the predicted output of a data point and its actual observed output. The lower the residual, the better the model. However, increasing the complexity of the model, that is, the number of degrees of freedom, will almost consistently decrease the residual error, since the model is allowed to describe more precisely not only the data but also the noise. Whereas the iterations in the optimization problem for a fixed size model can be continued until convergence, another criterion must be found to terminate the search for the optimal model complexity. We need a measure of the model relevance.

Cross-Validation

Cross-validation is a statistical practice that consists of splitting the data set available into different subsets, a training set and a test set. The former is used to define a fixed complexity model, while the latter serves to test it. The models are evaluated based on their capacity to describe the data, by checking how well a given model generalizes to new data. The name 'training set' indicates that the method has its origin in statistical learning theory. As the model complexity increases, the error on the training set decreases since it is always easier to describe the data using more parameters. The validation error will decrease in the beginning, as long as the model is still tuning in, but will eventually stabilize, or even increase due for example to noise in the training set. This break point corresponds to the 'optimal' complexity level.

In a common form, named K-fold cross-validation, the data is partitioned into K subsets that are used each in its turn as validation set. The optimization and validation are thus repeated K times. The remaining $K-1$ sets form the training data. If we let K be the number of data points, we obtain 'leave-one-out cross-validation' (LOOCV). In order not to 'waste' too much data on the validation set, the LOOCV should be preferred. However, for large data sets and computationally expensive optimization processes, this method can be unfeasible and a smaller value on K must be chosen. In the case of linear least squares problems, closed-form solutions exist [69].

Cross-validation makes no prior assumption about the model or the distribution of the noise. This is important in the case of non physical models, where the residual errors are caused not only by the noise, but also by an insufficient model.

Minimum Description Length

Another family of methods is motivated by the Principle of Parsimony, or Occam's razor, that can be formulated approximately by 'All other things being equal, the simplest solution is the best'. The idea is thus to choose the model that gives the shortest description of data. The method has its origins in the algorithmic or descriptive complexity theory of Kolmogorov. The description length is a measure of the compression ratio that we obtain, using the model instead of the data. For completely random data, no (lossless) compression is possible, and no model smaller than the data set is able to reproduce it correctly. On the other hand, if a model is appropriate, then it can be used instead of the data to store the signal. Among the appropriate models, the one with the shortest code length representing the model parameters and the residual error is retained.

A Minimum Description Length (MDL) formulation for model selection was introduced by Rissanen in [54]. It consists in associating a cost with the quantity of information necessary to describe the curve. As opposed to cross-validation that is an iterative method for nonlinear problems, MDL yields a criterion that is to be minimized. Different criteria follow, depending on the formulation of the estimation problem.

AIC

In 1974, Akaike presented a criterion, known as the Akaike Information Criterion (AIC) [1], intended to compare two models of different complexity. Increasing the number of model parameters to be estimated improves the goodness of fit, regardless of the number of parameters needed to generate the data. In order to avoid over-fitting, the method combines a term measuring the goodness of fit and a term penalizing the model complexity. Given two estimated models, differentiated by their number of parameters, the one with the lowest criterion will be retained. The AIC is written

$$AIC = -2L_m + 2k, \tag{1.50}$$

where L_m is the maximized log likelihood of the model and k is the number of parameters of the model. The formulation is rather simple, in contrast to more traditional approaches starting from a null hypothesis.

As for MDL, AIC originates from information-theoretic concepts, and is derived using the Kullback-Leibler divergence. Grounded on the concept of entropy, it provides a relative measure of the information lost when a given model is used to describe data. The method approaches the model selection problem from the point of view of prediction. Asymptotically, as the amount of data goes to infinity, the model will be chosen that has the best likelihood for future data.

BIC

A similar criterion, based on a Bayesian formalism, is the Bayesian Information Criterion (BIC) presented in 1978 by Schwarz [56]. It stresses the number of data points n, so as to ensure an asymptotic consistency and is written,

$$BIC = -2L_m + k \ln n, \tag{1.51}$$

where L_m is the maximized log likelihood, k is the number of parameters and n is the number of data points. BIC penalizes the model complexity more strongly than does AIC, thus being more conservative. Asymptotically, as the amount of data goes to infinity, it will choose the model that was used to generate the data.

Considering the Laplace approximation, that replaces the posterior distribution with a Gaussian distribution centered at the Maximum A Priori (MAP) estimate, BIC can be derived by taking the large sample limit as the number of data points tends to infinity.

Comparison

So, which method should be preferred? The answer depends as expected on the context, such as the amount of data available and the computational complexity of the optimization method used to obtain a fixed size model. Cross-validation is better founded for non-physical models if we dispose of enough data, but it becomes unfeasible for expensive optimization methods. The methods based on different criteria, such as AIC or BIC, present the obvious advantage of using only the training error. Asymptotically, the use of AIC and LOOCV should be equivalent. For a judicious choice of K, the same is true about BIC and K-fold cross-validation.

AIC and BIC both assumes that the data distribution is exponential, whereas MDL avoids assumptions about the data generating process altogether. If the distribution is known not to be exponential, or the log likelihood is unknown, MDL should thus be preferred over AIC and BIC. Since the criterion induced by the MDL method depends strongly on the formulation of the estimation problem, it is hard to draw any conclusions about its performance in a general setting.

1.8 Visibility of the Object

In order to reconstruct an object based on image observations, we must first determine which parts are visible in the image at the actual position and orientation of the object. Even though all points of the model can be projected onto the image plane, not all of them will actually be visible, due to occlusions. We will handle only the case of self-occlusion, meaning that all other forms of occlusion will be treated as noise.

An intuitive way of dealing with visibility is to use a z-buffer, that is a buffer with the same dimensions as the image where the depth, relative to the camera, of the point shown is saved. The idea is to retain the depth information, so as to visualize for each pixel the image of the point of the model that is the closest to the camera. This algorithm demands quite a lot of memory and does not take advantage of the fact that the model is the same for the set of images. Moreover, for each pixel, all points of the model that are projected there will have to be investigated. A more efficient method is the Binary Space Partition (BSP) trees algorithm, which was introduced in 1980 by Fuchs et al. [23].

1.8.1 Binary Space Partition Trees

The idea of the BSP algorithm is to generate a binary tree that classes the three-dimensional space into convex regions. This classification of space is then used to limit the search. The generation of the tree is a rather time consuming operation, but since it need not be done more than once for each model it can be done off-line and the time is no longer a problem.

1.8.2 Creating the BSP Tree

Although the algorithm will be applied to the polygons of the model in space, we will here treat the simpler case of line segments in the plane. Note that with a curved model, we will use a triangulation of the CAD model to define the polygons and line segments. The method described generalizes without problem to the setting that interests us. Consider a set of (randomly numbered) 2D lines. Our goal is to partition the plane into convex regions such that each region contains at most one segment. This will be accomplished by choosing a set of lines splitting the plane. We will use auto-partitioning, that is the splitting lines will be taken to coincide with the line segments. We start by picking a segment, drawing the line passing through it and defining a positive and a negative side. For all other line segments one of the following will be true

1. The segment is entirely contained within the splitting line.
2. The segment is crossed by the splitting line.
3. The segment is contained in the positive half-space defined by the splitting line.
4. The segment is contained in the negative half-space defined by the splitting line.

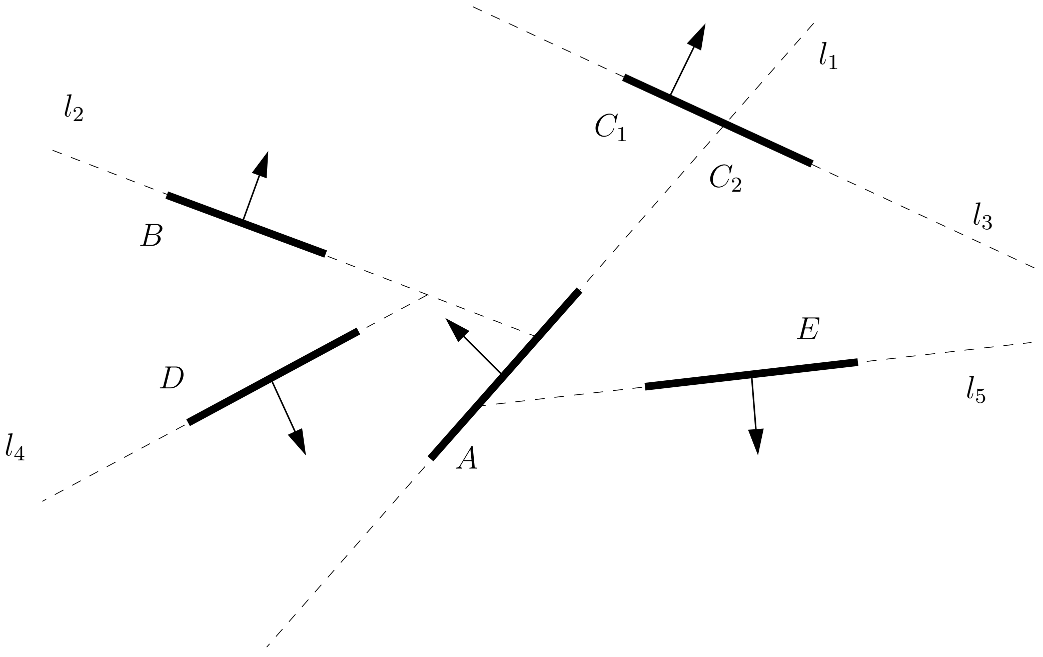
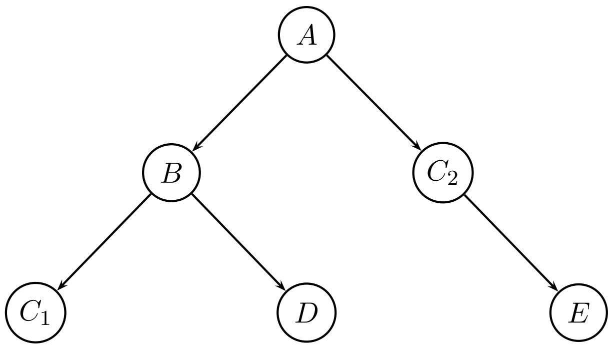


Figure 1.5: Example of a configuration of (ordered) lines in two-dimensional space and the chosen splitting lines l_i with positive directions indicated.



$$\begin{tikzcd}
 & & A \arrow[dl] \arrow[dr] & & \\
 & B \arrow[dl] \arrow[dr] & & C_2 \arrow[dr] & \\
C_1 & & D & & E
\end{tikzcd}$$

Figure 1.6: The resulting BSP tree from the line configuration in figure 1.5, where nodes in the left subtree belong to the positive half-space.

The segments from the first case together with the one used to define the splitting line are assembled in a list and forms a node in the tree. In the second case, the line segment is split into two new segments, one on the positive side and one on the negative side. Now proceed recursively by creating two new trees, one using the set of segments on the positive side and the other on the negative side. The splitting stops when there is at most one segment in the actual region. For an example of a configuration of (ordered) lines and the corresponding BSP tree, see figure 1.5 and figure 1.6 respectively.

Once the tree is constructed, every path from the root to a leaf describes a convex region of the plane. Note that different orderings of the segments result in different trees. The most efficient tree is of course a balanced one, since it allows for important cuts when used. A random ordering performs quite well and gives an expected number of final segments of $\mathcal{O}(n \log n)$ and an expected running time of $\mathcal{O}(n^2 \log n)$, where n is the number of initial segments.

The algorithm described generalizes to three dimensions by replacing the line segments with polygonal faces and the splitting lines with splitting planes.

1.8.3 Using the BSP Tree

Once the tree is created, we fix the position of the camera (given by the actual projection matrix) and the point for which we want to determine the visibility. The key observation is that if the view point and the point examined belong to the same half-space, everything in the other half-space will be 'behind' and need not be checked. In the BSP tree, the half-spaces correspond to the left and right subtrees. This leads to a cut in the tree that reduces the area to search through. A segment that belongs to the same region as the camera will thus be completely visible and no checking needs to be done. After cutting everything that can be cut, we check the remaining segments by tracing the ray from the camera to the point in question in order to see whether it crosses another segment, thus occluding the point, or not, leaving it visible.

Chapter 2

Reconstruction of 3D Points

This chapter deals with 3D reconstruction of a rigid scene based on observations in multiple images, a field called Structure-from-Motion. When it comes to metric reconstruction from point matches, the nonlinear method of bundle adjustment is predominant. It does however need a rather good initialization in order to converge to the global optimum. A faster, linear method is thus needed to compute an approximate solution. Affine reconstruction by factorization is one such method. In its original form, the factorization algorithm needs all points to be visible and matched in all views, since its matrix formulation does not handle missing data. We propose a method to align several reconstructions so as to obtain an approximated maximum likelihood solution to the problem. The method is extended via an Expectation-Maximization approach to handle missing data points.

The methods developed have been published in the proceedings of SCIA 2005 [4], EMMCVPR 2005 [41] and RFIA 2006 [42] respectively.

Reconstruction de points 3D

Ce chapitre traite de la reconstruction de points 3D basée sur des observations de ces points dans plusieurs images, un domaine appelé “Structure-from-Motion”. Pour la reconstruction métrique à partir de points appariés, la méthode prédominante est l’ajustement de faisceaux. Etant non linéaire, celle-ci nécessite une initialisation proche du résultat afin de converger sur le minimum global. Une méthode linéaire, plus rapide, est donc nécessaire pour fournir une solution initiale. La reconstruction affine par factorisation en est une. Dans sa forme d’origine, l’algorithme de factorisation nécessite que tous les points soient visibles et appariés dans toutes les vues, puisque la formulation matricielle ne permet pas de gérer de données manquantes. Nous proposons une méthode d’alignement de plusieurs reconstructions afin d’obtenir une estimation proche de la solution au maximum de vraisemblance. Cette méthode est étendue par une approche “Expectation-Maximization”, qui gère les données manquantes.

Les méthodes développées ont été publiées dans les actes de SCIA 2005 [4], EMMCVPR 2005 [41] et RFIA 2006 [42] respectivement.

- Introduction
- Préliminaires
  Présentation des notations et hypothèses.
- Alignement de reconstructions 3D affines
  Présentation de la méthode d’alignement qui donne une solution approchant celle du maximum de vraisemblance. La méthode est d’abord présentée dans le cas de deux ensembles de caméras, pour des ensembles de données complètes. Elle est ensuite étendue au cas des données manquantes et de plusieurs ensembles de caméras.
- Méthodes d’alignement
  Présentation d’autres méthodes classiques d’alignement : Une première méthode basée sur la minimisation d’une erreur de transfert non-symmétrique, puis une méthode basée sur la factorisation directe des points 3D reconstruits et finalement la méthode classique d’ajustement de faisceaux.
- Approximation de l’erreur de reprojection
  Justifications de l’approximation introduite dans notre méthode.
- Evaluation expérimentale
  Notre méthode est comparée aux autres méthodes présentées.
- Conclusions

2.1 Introduction

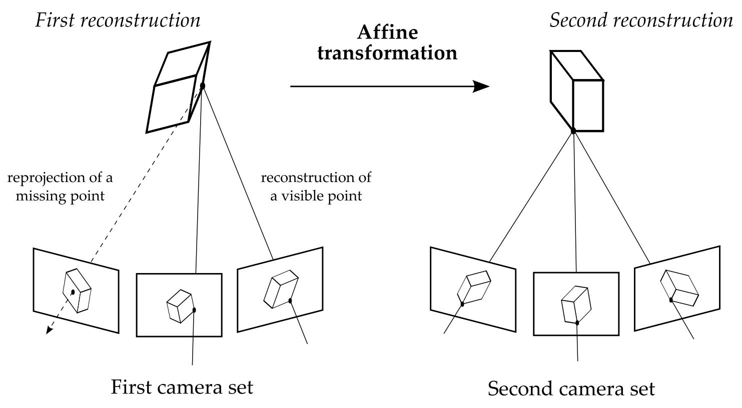


Figure 2.1: The problem tackled in this chapter is an approximation of the Maximum Likelihood Estimation of 3D affine transformations between two (or more) affine reconstructions obtained from uncalibrated affine cameras using an incomplete data set.

We focus on the uncalibrated affine camera model, which is a reasonable approximation to the perspective camera model when the depth of the observed scene is small compared to the viewing distance, see section 1.2.3 for details. In this case, the partial 3D models obtained from sub-sequences, that is, multiple subsets of cameras, are related by 3D affine transformations. We deal with the computation of such transformations from point correspondences, as illustrated on figure 2.1. We propose an approximation to a Maximum Likelihood Estimator based on factorizing modified image point coordinates. We compute a 3D affine transformation and a set of 3D point correspondences which perfectly match, such that the reprojection error in all sets of cameras is minimized. The method is intended to fit in hierarchical affine Structure-from-Motion processes of which the basic reconstruction block is, for example affine factorization [62]. We believe RANSAC [20] is the method of choice for reliable robust estimators to deal with data sets containing outliers. Embedding the proposed method in RANSAC is straightforward, since they work for both minimal and redundant data sets. The Maximum Likelihood estimate of the complete set of parameters can be obtained using bundle adjustment [65] over the 3D points, the affine transformation and the projection matrices, using the results from our method as an initialization. A final step would consist in performing a Euclidean reconstruction with simultaneous self-calibration of the cameras. Our method does not make any assumption about the cameras, besides the fact that a reconstruction of each camera set using an affine camera model has been performed. The method relies on the important new concept of *orthonormal bases*. In the occlusion-free case, our algorithm needs a single Singular Value Decomposition (SVD). However, in the case of incomplete measurement data, that is, when some of the 3D points used for the alignment are not visible in all views, the factorization algorithm must be extended. Though some algorithms have been proposed, see *e.g.* [33], they are not appropriate for Structure-from-Motion from large image sequences. As in [7, 26], we propose an Expectation-Maximization (EM) [14] based scheme. The Expectation step predicts the missing data while the Maximization step maximizes the log likelihood, that is, minimizes the reprojection error.

We proposed the approximation to the Maximum Likelihood Estimator in the complete data case in [4] and in the missing data case in [41].

This chapter is organized as follows. Our alignment method is described in Sect. 2.2, while other methods are summarized in Sect. 2.3. Experimental results are reported in Sect. 2.5. Our conclusions are given in Sect. 2.6.

2.2 Alignment of 3D Affine Reconstructions

We formally state the alignment problem in the two camera set case and present our algorithm. We first present the algorithm in the complete data case, dubbed 'FACTMLE', and then in the case with missing data, dubbed 'FACTMLE-EM'. Note that although our algorithms only approximates the true Maximum Likelihood error function, the approximation is so close that we use the term ML for naming our algorithms. Finally, we extend the method to the multiple camera set case.

2.2.1 Problem Statement

Given a set of point matches $\{\mathbf{x}_{ij}\}$, the factorization algorithm can be employed to recover all cameras $\{\hat{\mathsf{P}}_i, \hat{\mathbf{t}}_i\}$ and 3D points $\{\hat{\mathbf{Q}}_j\}$ at once [62]. An overview of the method, and more generally of Structure-from-Motion, is given in 1.5.

Consider two sets of cameras $\{(\mathsf{P}_i, \mathbf{t}_i)\}_{i=1}^{n}$ and $\{(\mathsf{P}'_i, \mathbf{t}'_i)\}_{i=1}^{n'}$ and the associated structures $\{\mathbf{Q}_j \leftrightarrow \mathbf{Q}'_j\}_{j=1}^{m}$ obtained by reconstructing a rigid scene using for example the above-mentioned factorization algorithm. Without loss of generality, we assume that the same number of points are present in the two reconstructions since only the point correspondences are used in the alignment task. The reprojection error over the two sets is given by

$$\mathcal{C}^2(\mathcal{Q}, \mathcal{Q}') = \mathcal{R}^2(\mathcal{P}, \mathcal{Q}, \{\mathbf{t}_i\}) + \mathcal{R}'^2(\mathcal{P}', \mathcal{Q}', \{\mathbf{t}'_i\}). \tag{2.1}$$

Letting $(\hat{\mathsf{A}}, \hat{\mathbf{t}})$ represent the aligning (3×3) affine transformation, the Maximum Likelihood Estimator is formulated by

$$\min_{\hat{\mathcal{Q}}, \hat{\mathcal{Q}}'} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \qquad \text{s.t.} \qquad \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}}. \tag{2.2}$$

2.2.2 A Factorization-Based Algorithm

Our method to solve problem (2.2) uses a three-step factorization strategy. We first describe it in the occlusion-free case and then propose an iterative extension for the missing data case.

Step 1: Orthonormalizing. We propose the important concept of *orthonormal bases.* We define a reconstruction to be in an orthonormal basis if the joint projection matrix is column-orthogonal. Given a joint projection matrix $\mathcal{P}$, one can find a 3D affine transformation represented by the (3×3) matrix N, which applies as A in equation (1.21), such that $\mathcal{P}\mathsf{N}$ is column-orthogonal, that is, such that $\mathsf{N}^\mathsf{T}\mathcal{P}^\mathsf{T}\mathcal{P}\mathsf{N} = \mathrm{I}_{(3\times3)}$. We call the transformation N an *orthonormalizing transformation.* The set of orthonormalizing transformations is 3-dimensional since for any 3D rotation matrix U, NU still is an orthonormalizing transformation for $\mathcal{P}$. We use the QR decomposition $\mathcal{P} = \mathsf{QR}$, see *e.g.* [25], giving an upper triangular orthonormalizing transformation $\mathsf{N} = \mathsf{R}^{-1}$. Other choices are possible for computing an N, *e.g.* if $\mathcal{P} = \mathsf{U\Sigma V}^\mathsf{T}$ is an SVD of $\mathcal{P}$, then $\mathsf{N} = \mathsf{V\Sigma}^{-1}$ has the required property. Henceforth, we assume that all 3D models are expressed in orthonormal bases

$$\left\{ \begin{array}{lcl} \mathcal{P} & \leftarrow & \mathcal{P}\mathsf{N} \\ \mathcal{P}' & \leftarrow & \mathcal{P}'\mathsf{N}' \end{array} \right. \qquad \text{and} \qquad \left\{ \begin{array}{lcl} \mathcal{Q} & \leftarrow & \mathsf{N}^{-1}\mathcal{Q} \\ \mathcal{Q}' & \leftarrow & \mathsf{N}'^{-1}\mathcal{Q}' \end{array} \right. .$$

An interesting property of orthonormal bases is that $\mathcal{P}^\dagger = \mathcal{P}^\mathsf{T}$. Hence, triangulating points in these bases is simply done by $\mathcal{Q} = \mathcal{P}^\mathsf{T}\mathcal{X}$.

Note that the matrix $\mathcal{P}$ computed by factorization, see Sect. 1.5, may already satisfy $\mathcal{P}^\mathsf{T}\mathcal{P} = \mathrm{I}$. However, if at least one of the cameras is not used for the alignment, *e.g.* if none of the 3D point correspondences project in this camera, or if the cameras come as the result of the alignment of partial 3D models, then $\mathcal{P}$ will *not* satisfy $\mathcal{P}^\mathsf{T}\mathcal{P} = \mathrm{I}$, thus requiring the orthonormalization step.

Step 2: Eliminating the Translation. The translation part of the sought-after transformation can not be computed directly, but can be eliminated from the equations. First, center the image points to eliminate the translation part of the cameras: $\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \mathbf{t}_i$ and $\mathbf{x}'_{ij} \leftarrow \mathbf{x}'_{ij} - \mathbf{t}'_i$. Second, consider that the partial derivatives of the reprojection error (2.1) with respect to $\hat{\mathbf{t}}$ must vanish $\frac{\partial \mathcal{C}^2}{\partial \hat{\mathbf{t}}} = 0$. By using the constraint $\hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}}$ from equation (2.2) and expanding using equation (2.1), we get

$$\begin{aligned}
\textstyle\sum_{i=1}^{n'}\sum_{j=1}^{m}\left(\mathsf{P}_i'^{\mathsf{T}}\mathsf{P}_i'\hat{\mathbf{t}} - \mathsf{P}_i'^{\mathsf{T}}\mathbf{x}'_{ij} + \mathsf{P}_i'^{\mathsf{T}}\mathsf{P}'_i\hat{\mathsf{A}}\hat{\mathbf{Q}}_j\right) &= 0\\
\textstyle\sum_{j=1}^{m}\left(\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathbf{t}} - \mathcal{P}'^{\mathsf{T}}\mathcal{Y}'_j + \mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\hat{\mathbf{Q}}_j\right) &= 0\\
m\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathbf{t}} - m\mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' + m\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\bar{\hat{\mathcal{Q}}} &= 0,
\end{aligned}$$

which yields $\hat{\mathbf{t}} = \left(\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\right)^{-1}(\mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' - \mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\bar{\hat{\mathcal{Q}}})$ that further simplifies to

$$\hat{\mathbf{t}} = \mathcal{P}'^{\dagger}\bar{\mathcal{Y}}' - \hat{\mathsf{A}}\bar{\hat{\mathcal{Q}}},$$

and, thanks to the orthonormal basis property $\mathcal{P}'^{\dagger} = \mathcal{P}'^{\mathsf{T}}$, we get

$$\hat{\mathbf{t}} = \mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' - \hat{\mathsf{A}}\bar{\hat{\mathcal{Q}}}, \tag{2.3}$$

Note that if the same entire sets of reconstructed points are used for the alignment, then we directly obtain $\hat{\mathbf{t}} = \mathbf{0}$ since $\bar{\mathcal{Y}}' = \mathbf{0}$ and $\bar{\hat{\mathcal{Q}}} = \mathbf{0}$. This is rarely the case in practice, especially if the alignment is used to merge partial 3D models.

Third, consider that the m partial derivatives of the reprojection error (2.1) with respect to each $\hat{\mathbf{Q}}_j$ must vanish as well: $\frac{\partial \mathcal{C}^2}{\partial \hat{\mathbf{Q}}_j} = 0$, and expand as above

$$\begin{aligned}
\textstyle\sum_{i=1}^{n}\left(\mathsf{P}_i^{\mathsf{T}}\mathsf{P}_i\hat{\mathbf{Q}}_j - \mathsf{P}_i^{\mathsf{T}}\mathbf{x}_{ij}\right) + \sum_{i=1}^{n'}\left(\hat{\mathsf{A}}^{\mathsf{T}}\mathsf{P}_i'^{\mathsf{T}}\mathsf{P}'_i\hat{\mathsf{A}}\hat{\mathbf{Q}}_j - \hat{\mathsf{A}}^{\mathsf{T}}\mathsf{P}_i'^{\mathsf{T}}\mathbf{x}'_{ij} + \hat{\mathsf{A}}^{\mathsf{T}}\mathsf{P}_i'^{\mathsf{T}}\mathsf{P}'_i\hat{\mathbf{t}}\right) &= 0\\
\mathcal{P}^{\mathsf{T}}\mathcal{P}\hat{\mathbf{Q}}_j - \mathcal{P}^{\mathsf{T}}\mathcal{Y}_j + \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\hat{\mathbf{Q}}_j - \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{Y}'_j + \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathbf{t}} &= 0.
\end{aligned}$$

The sum over j of all these derivatives also vanishes

$$\left(\forall j, \frac{\partial \mathcal{C}^2}{\partial \mathbf{Q}_j} = 0\right) \implies \left(\sum_{j=1}^{m}\frac{\partial \mathcal{C}^2}{\partial \mathbf{Q}_j} = 0\right),$$

giving

$$\mathcal{P}^{\mathsf{T}}\mathcal{P}\bar{\hat{\mathcal{Q}}} - \mathcal{P}^{\mathsf{T}}\bar{\mathcal{Y}} + \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\bar{\hat{\mathcal{Q}}} - \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' + \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathbf{t}} = 0.$$

By replacing $\hat{\mathbf{t}}$ by its expression (2.3), and after some minor algebraic manipulations, we obtain

$$\begin{aligned} \mathcal{P}^\mathsf{T}\mathcal{P}\bar{\hat{\mathcal{Q}}} - \mathcal{P}^\mathsf{T}\bar{\mathcal{Y}} &= 0 \\ \bar{\hat{\mathcal{Q}}} &= \mathcal{P}^\dagger\bar{\mathcal{Y}}, \end{aligned}$$

and by substituting in equation (2.3) and using the orthonormal basis property $\mathcal{P}^\dagger = \mathcal{P}^\mathsf{T}$, we get

$$\hat{\mathbf{t}} = \mathcal{P}'^\mathsf{T}\bar{\mathcal{Y}}' - \hat{\mathsf{A}}\mathcal{P}^\mathsf{T}\bar{\mathcal{Y}}. \tag{2.4}$$

It is common in factorization methods to center the data with respect to their centroid to cancel the translation part of the transformation. Equation (2.4) means that, according to the reprojection error criterion, the data must be centered with respect to the *reconstructed centroid* of the image points, not with respect to the actual 3D centroid.

Obviously, if the 3D models have been obtained by the factorization method of Sect. 1.5, then the centroid of the 3D points corresponds to the reconstructed centroid, that is, $\bar{\mathbf{Q}} = \mathcal{P}^\mathsf{T}\bar{\mathcal{Y}}$ and $\bar{\mathbf{Q}}' = \mathcal{P}'^\mathsf{T}\bar{\mathcal{Y}}'$, provided that the same sets of views are used for reconstruction and alignment.

To summarize, we cancel the translation part out of the sought-after transformation by translating the reconstructions and the image points as shown below

$$\left\{ \begin{array}{lcl} \mathbf{Q}_j & \leftarrow & \mathbf{Q}_j - \mathcal{P}^\mathsf{T}\bar{\mathcal{Y}} \\ \mathbf{Q}'_j & \leftarrow & \mathbf{Q}'_j - \mathcal{P}'^\mathsf{T}\bar{\mathcal{Y}}' \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{lcl} \mathbf{x}_{ij} & \leftarrow & \mathbf{x}_{ij} - \mathsf{P}_i\mathcal{P}^\mathsf{T}\bar{\mathcal{Y}} \\ \mathbf{x}'_{ij} & \leftarrow & \mathbf{x}'_{ij} - \mathsf{P}'_i\mathcal{P}'^\mathsf{T}\bar{\mathcal{Y}}'. \end{array} \right.$$

The reprojection error (2.1) is rewritten

$$\mathcal{C}^2(\mathcal{Q}, \mathcal{Q}') = \frac{1}{2nm}\left(\|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2 + \|\mathcal{X}' - \mathcal{P}'\mathcal{Q}'\|^2\right) \tag{2.5}$$

and problem (2.2) is reformulated as

$$\min_{\hat{\mathcal{Q}}, \hat{\mathcal{Q}}'} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \qquad \text{s.t.} \qquad \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j. \tag{2.6}$$

Step 3: Factorizing. Thanks to the orthonormal basis property $\mathcal{P}^\mathsf{T}\mathcal{P} = \mathrm{I}$, we can approximate the reprojection error on a single set of cameras as

$$\mathcal{R}^2(\mathcal{P}, \mathcal{Q}) \propto \|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2 \approx \|\mathcal{P}^\mathsf{T}\mathcal{X} - \mathcal{Q}\|^2.$$

We note that multiplication by $\mathcal{P}^\mathsf{T}$ corresponds to a projection onto the three-dimensional subspace spanned by the columns of $\mathcal{P}$. Since $\mathcal{X}$ already has rank 3, up to the measurement noise, the approximation is justified. For a discussion about the approximation of the reprojection error, we refer to appendix 2.4.

This allows us to rewrite the reprojection error (2.5) as

$$\widetilde{\mathcal{C}}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \propto \|\mathcal{P}^\mathsf{T}\mathcal{X} - \hat{\mathcal{Q}}\|^2 + \|\mathcal{P}'^\mathsf{T}\mathcal{X}' - \hat{\mathcal{Q}}'\|^2 = \left\| \underbrace{\begin{bmatrix} \mathcal{P}^\mathsf{T}\mathcal{X} \\ \mathcal{P}'^\mathsf{T}\mathcal{X}' \end{bmatrix}}_{\Lambda} - \underbrace{\begin{bmatrix} \hat{\mathcal{Q}} \\ \hat{\mathcal{Q}}' \end{bmatrix}}_{\Delta} \right\|^2. \quad (2.7)$$

By introducing the constraint $\hat{\mathcal{Q}}' = \hat{\mathsf{A}}\hat{\mathcal{Q}}$ from (2.6) and, as in Sect. 1.5, an unknown global affine transformation B

$$\Delta = \begin{bmatrix} \mathsf{I} \\ \hat{\mathsf{A}} \end{bmatrix} \mathsf{B}\mathsf{B}^{-1}\hat{\mathcal{Q}} = \underbrace{\begin{bmatrix} \mathsf{B} \\ \hat{\mathsf{A}}\mathsf{B} \end{bmatrix}}_{\tilde{\mathcal{M}}} \underbrace{\mathsf{B}^{-1}\mathcal{Q}}_{\tilde{\mathcal{Q}}}.$$

The problem is reformulated as

$$\min_{\tilde{\mathcal{M}}, \tilde{\mathcal{Q}}} \|\Lambda - \tilde{\mathcal{M}}\tilde{\mathcal{Q}}\|^2.$$

A solution is given by SVD of matrix Λ

$$\Lambda_{(6\times m)} = \mathsf{U}_{(6\times 6)}\Sigma_{(6\times 6)}\mathsf{V}^\mathsf{T}_{(6\times m)}.$$

Let $\Sigma = \Sigma_u\Sigma_v$ be any decomposition of matrix Σ. We obtain $\tilde{\mathcal{M}} = \psi(\mathsf{U}\Sigma_u)$ and $\tilde{\mathcal{Q}} = \psi^\mathsf{T}(\mathsf{V}\Sigma_v^\mathsf{T})$. Using the partitioning $\tilde{\mathcal{M}} = \begin{bmatrix} \tilde{\mathsf{M}} \\ \tilde{\mathsf{M}}' \end{bmatrix}$, we get

$$\left\{ \begin{array}{rcl} \mathsf{B} & = & \tilde{\mathsf{M}} \\ \hat{\mathsf{A}} & = & \tilde{\mathsf{M}}'\mathsf{B}^{-1} \\ \hat{\mathcal{Q}} & = & \mathsf{B}\tilde{\mathcal{Q}}. \end{array} \right.$$

Obviously, one needs to undo the effect of the orthonormalizing transformations, as follows

$$\left\{ \begin{array}{rcl} \hat{\mathsf{A}} & \leftarrow & \mathsf{N}'\hat{\mathsf{A}}\mathsf{N}^{-1} \\ \hat{\mathcal{Q}} & \leftarrow & \mathsf{N}\hat{\mathcal{Q}}. \end{array} \right.$$

This algorithm runs with $m \geq 4$ point correspondences[1]. Table 2.1 gives a summary.

[1]This is consistent with the fact that 4 point correspondences define a 3D affine transformation.

2.2.3 An Alternative Solution Method

Note that it is possible to solve the problem without using the orthonormalizing transformations, thus without the approximation introduced in (2.7). This solution requires however to compute the SVD of a $(2(n+n') \times m)$ matrix, made by stacking the measurement matrices $\mathcal{X}$ and $\mathcal{X}'$, and is therefore much more computationally expensive than the algorithm above, and may be intractable for large sets of cameras and points. We rewrite the cost function as

$$\mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') = \left\| \begin{bmatrix} \mathcal{X} \\ \mathcal{X}' \end{bmatrix} - \begin{bmatrix} \mathcal{P} & \\ & \mathcal{P}' \end{bmatrix} \begin{bmatrix} \hat{\mathcal{Q}} \\ \hat{\mathcal{Q}}' \end{bmatrix} \right\|^2 = \left\| \underbrace{\begin{bmatrix} \mathcal{X} \\ \mathcal{X}' \end{bmatrix}}_{\mathsf{\Lambda}} - \underbrace{\underbrace{\begin{bmatrix} \mathcal{P} & \\ & \mathcal{P}' \end{bmatrix}}_{\mathsf{E}} \underbrace{\begin{bmatrix} \mathsf{B} \\ \mathsf{B}^{-1}\hat{\mathsf{A}} \end{bmatrix} \hat{\mathcal{Q}}}_{\mathsf{Z}}}_{\mathsf{\Delta}} \right\|^2 .$$

A rank 3 approximation of matrix $\mathsf{\Lambda}$ by SVD gives the solution for $\mathsf{\Delta}$. Matrix Z is then recovered using least squares as

$$\mathsf{Z} = \mathsf{E}^\dagger \mathsf{\Delta}.$$

In order to extract the structure matrix $\hat{\mathcal{Q}}$ and the affine transformation $\hat{\mathsf{A}}$, we proceed as described above. We dub this alternative method FullSVD.

However, this method yields just another approximation of the Maximum Likelihood estimate, due to its two stage minimization of the reprojection error (2.5). Directly solving (2.6) requires a nonlinear approach, a method that we dub TrueMLE. So as to obtain the Maximum Likelihood estimate of the complete set of parameters, we also need to re-estimate the projection matrices, $\mathcal{P}$ and $\mathcal{P}'$. Such an estimation is performed at a later stage using bundle adjustment, see Sect. 2.3.4.

2.2.4 Dealing with Missing Data

The missing data case arises when some of the 3D points used for the alignment are not visible in all views. We propose an EM (Expectation Maximization) based extension of our algorithm to handle this case.

The EM framework has already been used in the context of Structure-from-Motion. In [7], a closed-form Maximum Likelihood Estimation replaces the factorization algorithm to perform an affine triangulation of N views. The algorithm then uses EM to perform the affine reconstruction based on all observed data. An EM-based approach to multibody factorization with missing data that allows one to incorporate prior knowledge and an arbitrary noise covariance by using temporal coherence between the views is presented

Algorithm

1. **Compute the orthonormalizing transformations**:

$$\left[\cdots \mathsf{P}_i{}^{\mathsf{T}} \cdots\right]^{\mathsf{T}} \overset{\text{QR}}{=} \mathcal{P}\mathsf{N}^{-1} \qquad \text{and} \qquad \left[\cdots \mathsf{P}_i'{}^{\mathsf{T}} \cdots\right]^{\mathsf{T}} \overset{\text{QR}}{=} \mathcal{P}'\mathsf{N}'^{-1}$$

2. **Compute the reconstructed centroids**:

$$\mathbf{C} = \frac{\mathcal{P}^{\mathsf{T}}}{m} \sum_{j=1}^{m} \begin{bmatrix} \vdots \\ \mathbf{x}_{ij} - \mathbf{t}_i \\ \vdots \end{bmatrix} \qquad \text{and} \qquad \mathbf{C}' = \frac{\mathcal{P}'^{\mathsf{T}}}{m} \sum_{j=1}^{m} \begin{bmatrix} \vdots \\ \mathbf{x}'_{ij} - \mathbf{t}'_i \\ \vdots \end{bmatrix}$$

3. **Form the measurement matrices**:

$$\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \mathsf{P}_i\mathbf{C} \qquad \text{and} \qquad \mathbf{x}'_{ij} \leftarrow \mathbf{x}'_{ij} - \mathsf{P}'_i\mathbf{C}'$$

4. **Factorize**:

$$\begin{bmatrix} \mathcal{P}^{\mathsf{T}}\mathcal{X} \\ \mathcal{P}'^{\mathsf{T}}\mathcal{X}' \end{bmatrix} \overset{\text{SVD}}{=} \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}} \quad \text{and set} \quad \begin{bmatrix} \tilde{\mathsf{M}} \\ \tilde{\mathsf{M}}' \end{bmatrix} = \psi(\mathsf{U}\sqrt{\Sigma}) \quad \text{and} \quad \tilde{\mathcal{Q}} = \psi^{\mathsf{T}}(\mathsf{V}\sqrt{\Sigma}).$$

5. **Recover the transformation**: Set $\hat{\mathsf{A}} = \mathsf{N}'\tilde{\mathsf{M}}'\tilde{\mathsf{M}}^{-1}\mathsf{N}^{-1}$ and $\mathbf{t} = \mathbf{C}' - \hat{\mathsf{A}}\mathbf{C}$.

6. **Recover the corrected points**: Set $\hat{\mathcal{Q}} = \mathsf{N}\tilde{\mathsf{M}}\tilde{\mathcal{Q}}$ and $\hat{\mathcal{Q}}' = \mathsf{N}'\tilde{\mathsf{M}}'\tilde{\mathcal{Q}}$.

7. **Transfer the points to the original coordinate frames**: Extract the corrected points $\hat{\mathbf{Q}}_j$ from $\hat{\mathcal{Q}}$ and $\hat{\mathbf{Q}}'_j$ from $\hat{\mathcal{Q}}'$. Translate them as $\hat{\mathbf{Q}}_j \leftarrow \hat{\mathbf{Q}}_j + \mathbf{C}$ and $\hat{\mathbf{Q}}'_j \leftarrow \hat{\mathbf{Q}}'_j + \mathbf{C}'$.

8. **Compute the reprojection error**:

$$\mathcal{C}^2(\mathcal{Q}, \mathcal{Q}') = \frac{1}{m} \sum_{j=1}^{m} \left(\frac{1}{n} \sum_{i=1}^{n} d^2(\mathbf{x}_{ij}, \mathsf{P}_i\hat{\mathbf{Q}}_j) + \frac{1}{n'} \sum_{i=1}^{n'} d^2(\mathbf{x}'_{ij}, \mathsf{P}'_i\hat{\mathbf{Q}}'_j) \right).$$

Table 2.1: FactMLE, the proposed approximated Maximum Likelihood alignment algorithm in the case of complete data.

in [26]. Another application to Structure-from-Motion of the EM algorithm is presented in [13], where it is used to learn simultaneously the 3D model and the data association. The missing information in this case is thus not the point coordinates themselves but the point correspondences between images.

The EM algorithm is an iterative method which estimates the model parameters, given an incomplete set of measurement data. The main idea is to alternate between predicting the missing data and estimating the model. Since the log likelihood cannot be maximized using factorization, due to the missing data, it is replaced by its conditional expectation given the observed data, using the current estimate of the parameters. The conditional expectation at iteration k is written

$$E^{(k)}(\Theta) = E\,[\,\log L(\Theta)\,|\,\mathcal{X}_{\text{obs}}, \Theta^{(k)}\,]\,, \tag{2.8}$$

where Θ is the set of parameters, $\log L(\Theta)$ the log likelihood and $\mathcal{X}_{\text{obs}}$ the observed data. In the maximization step, this conditional expectation is then maximized with respect to the parameters and the new parameter vector is given by

$$\Theta^{(k+1)} = \arg\max_{\Theta} E^{(k)}(\Theta). \tag{2.9}$$

In the case where the log likelihood is a linear function of the missing data, this simply consists in replacing the missing data by their conditional expectations given the observed data at current parameter values. This approximated log likelihood is then maximized so as to yield a new estimate of the parameters. The log likelihood increases in each iteration and the process converges to a local minimum of the approximated Maximum Likelihood residual error (2.1). The rate of convergence of the EM algorithm is linear and a function of the rate of missing data. See *e.g.* [46] for details and proof of convergence.

Since the reconstruction of the two camera sets using factorization needs a complete data set, we are limited to the points visible in all views for the initial reconstruction. This allows us to reconstruct all cameras, but only part of the 3D points. We then triangulate the missing points in order to complete the 3D point cloud. This preliminary expectation step yields a completed set of 3D data, that can be used in the alignment algorithm.

However, the reprojection error, that is, the negative log likelihood, still cannot be minimized because of the incomplete measurement matrix $\mathcal{X}$. The expectation step predicts the missing image points by reprojecting them from the completed 3D points according to

$$\left\{ \begin{array}{lcl} \mathcal{X}^{(k)} & = & E\,[\,\mathcal{X}\,|\,\mathcal{X}_{\text{obs}}, \hat{\mathcal{Q}}^{(k)}\,] \\ \mathcal{X}'^{(k)} & = & E\,[\,\mathcal{X}'\,|\,\mathcal{X}'_{\text{obs}}, \hat{\mathcal{Q}}'^{(k)}\,]. \end{array} \right. \tag{2.10}$$

Namely for the missing point $\mathbf{x}_{ij}$, we set

$$\mathbf{x}_{ij}^{(k+1)} \leftarrow \mathsf{P}_i^{(k)} \hat{\mathbf{Q}}_j^{(k)} + \mathbf{t}_i^{(k)}.$$

The maximization step consists in applying the algorithm described in the complete data case using the formulation

$$\min_{\hat{\mathcal{Q}},\hat{\mathcal{Q}}'} \frac{1}{2nm} \left(\|\mathcal{X}^{(k)} - \mathcal{P}\hat{\mathcal{Q}}\|^2 + \|\mathcal{X}'^{(k)} - \mathcal{P}'\hat{\mathcal{Q}}'\|^2 \right), \tag{2.11}$$

with $\hat{\mathcal{Q}}$ and $\hat{\mathcal{Q}}'$ such that $\hat{\mathcal{Q}}' = \hat{\mathsf{A}}\hat{\mathcal{Q}}$. This yields an estimate of the sought-after transformation $(\hat{\mathsf{A}}, \hat{\mathbf{t}})$ as well as corrected point positions $\{\hat{\mathbf{Q}}_j \leftrightarrow \hat{\mathbf{Q}}'_j\}$.

These two steps are alternated, thus forming an iterative procedure where the corrected points are used in the expectation at the next iteration. In order to decide whether convergence is reached, the change in reprojection error between two iterations is measured. When the reprojection error stabilizes, the final result is returned.

Table 2.2 gives a summary of the algorithm with its EM extension.

2.2.5 Using Multiple Camera Sets

Suppose that one is given l sets of reconstructed point correspondences $\{\mathbf{Q}_j \leftrightarrow \mathbf{Q}'_j \leftrightarrow \mathbf{Q}''_j \leftrightarrow \ldots\}$ and cameras $\{(\mathsf{P}_i, \mathbf{t}_i)\}, \{(\mathsf{P}'_i, \mathbf{t}'_i)\}, \{(\mathsf{P}''_i, \mathbf{t}''_i)\}, \ldots$. We show that our algorithm can be extended to compute all the aligning transformations $\{(\hat{\mathsf{A}}, \hat{\mathbf{t}}), (\hat{\mathsf{A}}', \hat{\mathbf{t}}'), \ldots\}$ and corrected point position $\{\hat{\mathbf{Q}}_j \leftrightarrow \hat{\mathbf{Q}}'_j \leftrightarrow \hat{\mathbf{Q}}''_j \leftrightarrow \ldots\}$ in one single computation step. The derivation of this algorithm is very similar to the two camera set case, so we shall not present it in great details.

The problem is formulated as

$$\min_{\hat{\mathcal{Q}},\hat{\mathcal{Q}}',\hat{\mathcal{Q}}'',\ldots} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}', \hat{\mathcal{Q}}'', \ldots) \qquad \text{s.t.} \qquad \left\{ \begin{array}{l} \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}} \\ \hat{\mathbf{Q}}''_j = \hat{\mathsf{A}}'\hat{\mathbf{Q}}_j + \hat{\mathbf{t}}' \\ \qquad \vdots \end{array} \right. \tag{2.12}$$

where $\mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}', \hat{\mathcal{Q}}'', \ldots)$ is the total reprojection error given by

$$\mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}', \hat{\mathcal{Q}}'', \ldots) = \frac{1}{lnm} \Big(\mathcal{R}^2(\mathcal{P}, \hat{\mathcal{Q}}, \{\mathbf{t}_i\}) + \\ + \mathcal{R}^2(\mathcal{P}', \hat{\mathcal{Q}}', \{\mathbf{t}'_i\}) + \mathcal{R}^2(\mathcal{P}'', \hat{\mathcal{Q}}'', \{\mathbf{t}''_i\}) + \ldots \Big). \tag{2.13}$$

<u>Algorithm</u>

1. **Compute the orthonormalizing transformations**:
$$\left[\cdots \mathsf{P}_i^{\mathsf{T}} \cdots\right]^{\mathsf{T}} \overset{\text{QR}}{=} \mathcal{P}\mathsf{N}^{-1} \qquad \text{and} \qquad \left[\cdots \mathsf{P}_i'^{\mathsf{T}} \cdots\right]^{\mathsf{T}} \overset{\text{QR}}{=} \mathcal{P}'\mathsf{N}'^{-1}$$
2. **Form the measurement matrices**: $\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \mathbf{t}_i$ and $\mathbf{x}'_{ij} \leftarrow \mathbf{x}'_{ij} - \mathbf{t}'_i$
3. **Expectation-Maximization**:
   (a) **Expectation**. Predict the missing point $\mathbf{x}_{ij}$ by setting $\mathbf{x}_{ij} \leftarrow \mathsf{P}_i\hat{\mathbf{Q}}_j$. Compute the reconstructed centroids:
$$\mathbf{C} = \frac{\mathcal{P}^{\mathsf{T}}}{m}\sum_{j=1}^{m}\begin{bmatrix}\vdots\\ \mathbf{x}_{ij}\\ \vdots\end{bmatrix} \qquad \text{and} \qquad \mathbf{C}' = \frac{\mathcal{P}'^{\mathsf{T}}}{m}\sum_{j=1}^{m}\begin{bmatrix}\vdots\\ \mathbf{x}'_{ij}\\ \vdots\end{bmatrix}$$
   Recenter the measurement matrices:
$$\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \mathsf{P}_i\mathbf{C} \qquad \text{and} \qquad \mathbf{x}'_{ij} \leftarrow \mathbf{x}'_{ij} - \mathsf{P}'_i\mathbf{C}'.$$
   (b) **Maximization**. Factorize:
$$\begin{bmatrix}\mathcal{P}^{\mathsf{T}}\mathcal{X}\\ \mathcal{P}'^{\mathsf{T}}\mathcal{X}'\end{bmatrix} \overset{\text{SVD}}{=} \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}} \quad \text{and set} \quad \begin{bmatrix}\tilde{\mathsf{M}}\\ \tilde{\mathsf{M}}'\end{bmatrix} = \psi(\mathsf{U}\sqrt{\Sigma}) \quad \text{and} \quad \tilde{\mathcal{Q}} = \psi^{\mathsf{T}}(\mathsf{V}\sqrt{\Sigma}).$$
   (c) **Recover the corrected points**. Set $\hat{\mathcal{Q}} = \mathsf{N}\tilde{\mathsf{M}}\tilde{\mathcal{Q}}$ and $\hat{\mathcal{Q}}' = \mathsf{N}'\tilde{\mathsf{M}}'\tilde{\mathcal{Q}}$.
   (d) **Transfer the points to the original coordinate frames**. Extract the corrected points $\hat{\mathbf{Q}}_j$ from $\hat{\mathcal{Q}}$. Translate them as $\hat{\mathbf{Q}}_j \leftarrow \hat{\mathbf{Q}}_j + \mathbf{C}$.
   (e) **Compute the reprojection error**.
$$\mathcal{C}^2(\mathcal{Q},\mathcal{Q}') = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{n}\sum_{i=1}^{n} d^2(\mathbf{x}_{ij},\mathsf{P}_i\hat{\mathbf{Q}}_j) + \frac{1}{n'}\sum_{i=1}^{n'} d^2(\mathbf{x}'_{ij},\mathsf{P}'_i\hat{\mathbf{Q}}'_j)\right).$$
   (f) **Loop on** (a). Iterate until convergence is reached (see Sect. 2.2.4).
4. **Recover the transformation**: Set $\hat{\mathsf{A}} = \mathsf{N}'\tilde{\mathsf{M}}'\tilde{\mathsf{M}}^{-1}\mathsf{N}^{-1}$ and $\mathbf{t} = \mathbf{C}' - \hat{\mathsf{A}}\mathbf{C}$.

Table 2.2: FactMLE-EM, the proposed approximated Maximum Likelihood alignment algorithm with its EM extension handling the missing data case.

Step 1: Orthonormalizing. Following Sect. 2.2.2, we introduce the orthonormalizing transformations $\mathsf{N}, \mathsf{N}', \mathsf{N}'', \dots$ and apply them to all reconstructions

$$\begin{cases} \mathcal{P} & \leftarrow & \mathcal{P}\mathsf{N} \\ \mathcal{P}' & \leftarrow & \mathcal{P}'\mathsf{N}' \\ \mathcal{P}'' & \leftarrow & \mathcal{P}''\mathsf{N}'' \\ & \vdots & \end{cases} \quad \text{and} \quad \begin{cases} \mathcal{Q} & \leftarrow & \mathsf{N}^{-1}\mathcal{Q} \\ \mathcal{Q}' & \leftarrow & \mathsf{N}'^{-1}\mathcal{Q}' \\ \mathcal{Q}'' & \leftarrow & \mathsf{N}''^{-1}\mathcal{Q}'' \\ & \vdots & \end{cases} .$$

Step 2: Eliminating the Translations. We eliminate the translation parts $\{\hat{\mathbf{t}}, \hat{\mathbf{t}}', \dots\}$ of the 3D transformations. Briefly, we compute the partial derivatives of the reprojection error (2.13) with respect to each translation. By substituting the constraints in (2.12) and nullifying the resulting equations, expressions similar to (2.3) are obtained. Likewise, nullifying the partial derivatives with respect to the corrected points $\{\hat{\mathbf{Q}}_j\}$, and substituting the expressions for the translations, we obtain equation (2.2.2), $\bar{\hat{\mathcal{Q}}} = \mathcal{P}^\dagger \bar{\mathcal{Y}}$, leading, using the orthonormal bases property $\mathcal{P}^\dagger = \mathcal{P}^\mathsf{T}$, to

$$\begin{aligned} \hat{\mathbf{t}} &= \mathcal{P}'^\mathsf{T}\bar{\mathcal{Y}}' - \hat{\mathsf{A}}\mathcal{P}^\mathsf{T}\bar{\mathcal{Y}} \\ \hat{\mathbf{t}}' &= \mathcal{P}''^\mathsf{T}\bar{\mathcal{Y}}'' - \hat{\mathsf{A}}'\mathcal{P}^\mathsf{T}\bar{\mathcal{Y}} \\ &\vdots \end{aligned}$$

This means, as equation (2.4), that the centered coordinates canceling the translation parts out of the transformations are obtained by translating each set of 3D points so that the centroids reconstructed from the images lie at the origin. By translating the image points as in Sect. 2.2.2 to get rid of the translation part of the cameras, problem (2.12) can be expressed as

$$\min_{\hat{\mathcal{Q}}, \hat{\mathcal{Q}}', \hat{\mathcal{Q}}'', \dots} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}', \hat{\mathcal{Q}}'', \dots) \qquad \text{s.t.} \qquad \begin{cases} \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j \\ \hat{\mathbf{Q}}''_j = \hat{\mathsf{A}}'\hat{\mathbf{Q}}_j \\ \vdots \end{cases} \tag{2.14}$$

where

$$\begin{aligned} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}', \hat{\mathcal{Q}}'', \dots) &\propto \mathcal{R}^2(\mathcal{P}, \hat{\mathcal{Q}}) + \mathcal{R}^2(\mathcal{P}', \hat{\mathcal{Q}}') + \mathcal{R}^2(\mathcal{P}'', \hat{\mathcal{Q}}'') + \dots \\ &= \|\mathcal{X} - \mathcal{P}\hat{\mathcal{Q}}\|^2 + \|\mathcal{X}' - \mathcal{P}'\hat{\mathcal{Q}}'\|^2 + \|\mathcal{X}'' - \mathcal{P}''\hat{\mathcal{Q}}''\|^2 + \dots \end{aligned} \tag{2.15}$$

Step 3: Factorizing. Thanks to the orthonormal bases, we approximate the reprojection error (2.15) by

$$\tilde{\mathcal{C}}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}', \hat{\mathcal{Q}}'', \dots) = \left\| \underbrace{\begin{bmatrix} \mathcal{P}^\mathsf{T}\mathcal{X} \\ \mathcal{P}'^\mathsf{T}\mathcal{X}' \\ \mathcal{P}''^\mathsf{T}\mathcal{X}'' \\ \vdots \end{bmatrix}}_{\Lambda} - \underbrace{\begin{bmatrix} \hat{\mathcal{Q}} \\ \hat{\mathcal{Q}}' \\ \hat{\mathcal{Q}}'' \\ \vdots \end{bmatrix}}_{\Delta} \right\|^2 .$$

By introducing the constraints from equation (2.14) and an unknown global affine transformation B

$$\Delta = \begin{bmatrix} \mathrm{I} \\ \hat{\mathsf{A}} \\ \hat{\mathsf{A}}' \\ \vdots \end{bmatrix} \mathsf{B}\mathsf{B}^{-1}\hat{\mathcal{Q}} = \underbrace{\begin{bmatrix} \mathsf{B} \\ \hat{\mathsf{A}}\mathsf{B} \\ \hat{\mathsf{A}}'\mathsf{B} \\ \vdots \end{bmatrix}}_{\tilde{\mathcal{M}}} \underbrace{\mathsf{B}^{-1}\hat{\mathcal{Q}}}_{\tilde{\mathcal{Q}}} .$$

Let $l \geq 2$ be the number of camera sets. The SVD of matrix Λ is

$$\Lambda_{(3l\times m)} = \mathsf{U}_{(3l\times m)}\Sigma_{(m\times m)}\mathsf{V}^\mathsf{T}_{(m\times m)} .$$

Let $\Sigma = \Sigma_u\Sigma_v$ be any decomposition of matrix Σ, we obtain $\tilde{\mathcal{M}} = \psi(\mathsf{U}\Sigma_u)$ and $\tilde{\mathcal{Q}} = \psi^\mathsf{T}(\mathsf{V}\Sigma_v^\mathsf{T})$. By using the partitioning $\tilde{\mathcal{M}}^\mathsf{T} = (\tilde{\mathsf{M}}^\mathsf{T} \; \tilde{\mathsf{M}}'^\mathsf{T} \; \tilde{\mathsf{M}}''^\mathsf{T} \; \cdots)$ while undoing the orthonormalizing transformations, one gets

$$\left\{ \begin{array}{lcl} \mathsf{B} & = & \mathsf{N}\tilde{\mathsf{M}} \\ \hat{\mathsf{A}} & = & \mathsf{N}'\tilde{\mathsf{M}}'\mathsf{B}^{-1} \\ \hat{\mathsf{A}}' & = & \mathsf{N}''\tilde{\mathsf{M}}''\mathsf{B}^{-1} \\ & \vdots & \end{array} \right. \quad \text{and} \quad \left\{ \begin{array}{lcl} \hat{\mathcal{Q}} & = & \mathsf{B}\tilde{\mathcal{Q}} \\ \hat{\mathcal{Q}}' & = & \hat{\mathsf{A}}\mathsf{B}\tilde{\mathcal{Q}} \\ \hat{\mathcal{Q}}'' & = & \hat{\mathsf{A}}'\mathsf{B}\tilde{\mathcal{Q}} \\ & \vdots & \end{array} \right.$$

2.3 Other Algorithms

We briefly describe three other alignment algorithms. The first two of them do not yield Maximum Likelihood estimates under the previously-mentioned hypotheses on the noise distribution. They rely on 3D measurements and therefore naturally handle missing image data. The third method is the classic nonlinear method of bundle adjustment, giving the Maximum Likelihood estimate with respect to the parameters of the whole model, namely the projection and the structure matrices.

2.3.1 Minimizing the Non-Symmetric Transfer Error

This algorithm, dubbed 'TRERROR', is specific to the two camera set case. It is based on minimizing a non-symmetric 3D transfer error $\mathcal{E}(\hat{\mathsf{A}})$ as follows

$$\min_{\hat{\mathsf{A}},\hat{\mathbf{t}}} \mathcal{E}^2(\hat{\mathsf{A}}, \hat{\mathbf{t}}) \qquad \text{with} \qquad \mathcal{E}^2(\hat{\mathsf{A}}) = \frac{1}{m}\sum_{j=1}^{m} \|\mathbf{Q}'_j - \hat{\mathsf{A}}\mathbf{Q}_j - \hat{\mathbf{t}}\|^2.$$

Differentiating $\mathcal{E}^2$ with respect to $\hat{\mathbf{t}}$ and nullifying the result yields

$$\hat{\mathbf{t}} = \hat{\mathbf{Q}}' - \hat{\mathsf{A}}\hat{\mathbf{Q}}.$$

Henceforth, we assume that the translation has been eliminated by translating each 3D point set on its centroid. By rewriting the error function as

$$\mathcal{E}^2(\hat{\mathsf{A}}) \propto \|\mathcal{Q}' - \hat{\mathsf{A}}\mathcal{Q}\|^2,$$

and applying standard linear least-squares, one obtains the solution

$$\hat{\mathsf{A}} = \mathcal{Q}'\mathcal{Q}^{\dagger}.$$

2.3.2 Direct 3D Factorization

This algorithm, dubbed 'FACT3D', is based on directly factorizing the 3D reconstructed points. It is not restricted to the two camera set case, but for simplicity, we only describe this case. Generalization to multiple camera sets is trivial. The algorithm computes the aligning transformation $(\hat{\mathsf{A}}, \hat{\mathbf{t}})$ and perfectly corresponding points $\{\hat{\mathbf{Q}}_j \leftrightarrow \hat{\mathbf{Q}}'_j\}$. The reconstructed cameras are not taken into account by this algorithm, which entirely relies on 3D measurements on the reconstructed points. Under certain conditions, examined subsequently, this algorithm is equivalent to the proposed FACTMLE-EM.

The problem is stated as

$$\min_{\hat{\mathcal{Q}},\hat{\mathcal{Q}}'} \mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \qquad \text{s.t.} \qquad \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}},$$

where the 3D error function employed is defined by

$$\mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') = \frac{1}{2m}\left(\|\mathcal{Q} - \hat{\mathcal{Q}}\|^2 + \|\mathcal{Q}' - \hat{\mathcal{Q}}'\|^2\right). \qquad (2.16)$$

Minimizing this error function means that if the noise *on the 3D point coordinates* were Gaussian, centered and *i.i.d.*, which is *not* the case with our actual hypotheses, then this algorithm would yield the Maximum Likelihood estimate.

Step 1: Computing the Translation. By nullifying the partial derivatives of the error function $\mathcal{D}^2$ with respect to $\hat{\mathbf{t}}$ and with respect to the $\hat{\mathbf{Q}}_j$, and substituting the latter expressions into the former one, we obtain

$$\hat{\mathbf{t}} = \bar{\mathbf{Q}}' - \hat{\mathsf{A}}\bar{\mathbf{Q}}.$$

This equation means that, as in most factorization methods, canceling the translation part out according to the error function $\mathcal{D}$ is done by centering each set of 3D points on its actual centroid: $\hat{\mathbf{Q}}_j \leftarrow \hat{\mathbf{Q}}_j - \bar{\mathbf{Q}}$ and $\hat{\mathbf{Q}}'_j \leftarrow \hat{\mathbf{Q}}'_j - \bar{\mathbf{Q}}'$. Henceforth, we assume to work in centered coordinates. The problem is rewritten as

$$\min_{\hat{\mathcal{Q}},\hat{\mathcal{Q}}'} \mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \qquad \text{s.t.} \qquad \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j.$$

Step 2: Factorizing. Following the approach in Sect. 2.2.2, we rewrite $\mathcal{D}$ as

$$\mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \propto \left\| \begin{bmatrix} \mathcal{Q} \\ \mathcal{Q}' \end{bmatrix} - \begin{bmatrix} \hat{\mathcal{Q}} \\ \hat{\mathcal{Q}}' \end{bmatrix} \right\|^2 = \left\| \underbrace{\begin{bmatrix} \mathcal{Q} \\ \mathcal{Q}' \end{bmatrix}}_{\Lambda} - \underbrace{\begin{bmatrix} \mathsf{B} \\ \mathsf{AB} \end{bmatrix}}_{\tilde{\mathcal{M}}} \underbrace{\mathsf{B}^{-1}\hat{\mathcal{Q}}}_{\tilde{\mathcal{Q}}} \right\|^2.$$

Using SVD of matrix $\Lambda = \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$, we obtain $\tilde{\mathcal{M}} = \psi(\mathsf{U}\Sigma_u)$ and $\tilde{\mathcal{Q}} = \psi^\mathsf{T}(\mathsf{V}\Sigma_v^\mathsf{T})$. By using the partitioning $\tilde{\mathcal{M}} = \begin{bmatrix} \tilde{\mathsf{M}} \\ \tilde{\mathsf{M}}' \end{bmatrix}$, we get

$$\begin{cases} \mathsf{B} & = & \tilde{\mathsf{M}} \\ \hat{\mathsf{A}} & = & \tilde{\mathsf{M}}'\mathsf{B}^{-1} \\ \hat{\mathcal{Q}} & = & \mathsf{B}\tilde{\mathcal{Q}} \end{cases}.$$

Equivalence between FACT3D **and** FACTMLE. They are two conditions under which FACT3D minimizes a 2D error. It is then, in the complete data case, equivalent to the FACTMLE algorithm. Not surprisingly, the first one is that the 3D models must be expressed in orthonormal bases, that is, $\mathcal{P}^\mathsf{T}\mathcal{P} = \mathcal{P}'^\mathsf{T}\mathcal{P}' = \mathrm{I}$. The second one is that each 3D model must have been computed using Maximum Likelihood point triangulation, which imply in particular that all the views used for the reconstruction must be used for the alignment phase, that is, $\mathcal{Q} = \mathcal{P}^\mathsf{T}\mathcal{X}$ and $\mathcal{Q}' = \mathcal{P}'^\mathsf{T}\mathcal{X}'$. While the first condition is easy to satisfy by using an orthonormalizing transformation as described in Sect. 2.2.2, the second one is difficult to meet in practice. Indeed, when one reconstruct from image sequences, the alignment phase consists in registering partial 3D models using small subsets of points. Verifying that, under these two conditions, the approximated Maximum Likelihood residual error (2.1) can be rewritten as $\mathcal{D}$ is straightforward.

2.3.3 Computing the Reprojection Error

Each point correspondence contributes to the reprojection error. While for method FACTMLE-EM, the corrected points computed by the algorithm directly provide the reprojection error, for methods TRERROR and FACT3D, corrected points have to be estimated by minimizing the reprojection error. This minimization is conducted independently for each point. The reprojection error that we want to measure with respect to an estimated transformation $(\hat{\mathsf{A}}, \hat{\mathbf{t}})$ is defined by equation (2.1)

$$\mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{n}\sum_{i=1}^{n} d^2(\mathbf{x}_{ij}, \mathsf{P}_i\hat{\mathbf{Q}}_j + \mathbf{t}_i) + \frac{1}{n'}\sum_{i=1}^{n'} d^2(\mathbf{x}_{ij}, \mathsf{P}'_i\hat{\mathbf{Q}}'_j + \mathbf{t}'_i)\right).$$

Introduce the constraint $\hat{\mathcal{Q}}' = \hat{\mathsf{A}}\hat{\mathcal{Q}} + \hat{\mathbf{t}}$

$$\mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathsf{A}}\hat{\mathcal{Q}} + \hat{\mathbf{t}}) = \frac{1}{m}\sum_{j=1}^{m} e^2(\hat{\mathbf{Q}}_j, (\hat{\mathsf{A}}, \hat{\mathbf{t}})),$$

where

$$e^2(\hat{\mathbf{Q}}_j, (\hat{\mathsf{A}}, \hat{\mathbf{t}})) = \frac{1}{n}\sum_{i=1}^{n} d^2(\mathbf{x}_{ij}, \mathsf{P}_i\hat{\mathbf{Q}}_j + \mathbf{t}_i) + \frac{1}{n'}\sum_{i=1}^{n'} d^2(\mathbf{x}_{ij}, \mathsf{P}'_i(\hat{\mathsf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}}) + \mathbf{t}'_i).$$

The term $e^2(\hat{\mathbf{Q}}_j, (\hat{\mathsf{A}}, \hat{\mathbf{t}}))$ is the contribution of the j-th point correspondence to the reprojection error. It is given by

$$e^2\left(\arg\min_{\hat{\mathbf{Q}}_j} e^2(\hat{\mathbf{Q}}_j, (\hat{\mathsf{A}}, \hat{\mathbf{t}})), (\hat{\mathsf{A}}, \hat{\mathbf{t}})\right).$$

The inner minimization is a linear least-squares problem

$$e^2(\hat{\mathbf{Q}}_j, (\hat{\mathsf{A}}, \hat{\mathbf{t}})) = \left\| \begin{bmatrix} \vdots \\ \mathsf{P}_i \\ \vdots \\ \mathsf{P}'_i\hat{\mathsf{A}} \\ \vdots \end{bmatrix} \hat{\mathbf{Q}}_j - \begin{bmatrix} \vdots \\ \mathbf{x}_{ij} - \mathbf{t}_i \\ \vdots \\ \mathbf{x}'_{ij} - \mathbf{t}'_i - \mathsf{P}'_i\hat{\mathbf{t}} \\ \vdots \end{bmatrix} \right\|^2,$$

where the matrices are of size $(2(n+n') \times 3)$ and $(2(n+n') \times 1)$ respectively. We now solve the problem using a standard matrix pseudoinverse technique.

2.3.4 Bundle Adjustment

In order to obtain the Maximum Likelihood estimate of the complete set of parameters, we minimize the reprojection error from (2.1). The minimization is now carried out by re-estimating the structure as well as the projection matrices. The problem is stated as

$$\min_{\mathcal{P},\mathcal{P}',\hat{\mathcal{Q}},\hat{\mathcal{Q}}'} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \qquad \text{s.t.} \qquad \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}}. \tag{2.17}$$

Although the expression still is concise, the problem is not bilinear anymore and can thus not be solved using a matrix factorization technique. The nonlinear minimization, called bundle adjustment (Bundle), is commonly performed by using the Levenberg-Marquardt method, a method that finds the minimum of a sum of squares of nonlinear functions. Writing the reprojection error as

$$\mathcal{R}^2(\mathcal{P}, \mathcal{P}', \mathcal{Q}, \mathcal{Q}') = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{n}\sum_{i=1}^{n} d^2(\mathbf{x}_{ij}, \mathsf{P}_i\mathbf{Q}_j) + \frac{1}{n'}\sum_{i=1}^{n'} d^2(\mathbf{x}'_{ij}, \mathsf{P}'_i\mathbf{Q}'_j)\right), \tag{2.18}$$

we can use the technique presented in 1.5 to solve the minimization problem. In order to converge to the global minimum of the reprojection error, the method needs an initial value not too far from the solution to converge, which is provided by one of the linear methods presented above.

Note that since (2.18) does not use the matrix formulation that was necessary for the linear approach, the reprojection error is computed at all measured image points, thus avoiding the missing data problem.

2.4 An Approximation to the Reprojection Error

As the reprojection error will be used in an optimization process, we do not need to care about scale factors. We will therefore formulate the reprojection error using equality up to a constant ($\propto$). Whereas the minimization of the reprojection error (2.5),

$$\mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \propto \|\mathcal{X} - \mathcal{P}\hat{\mathcal{Q}}\|^2 + \|\mathcal{X}' - \mathcal{P}'\hat{\mathcal{Q}}'\|^2,$$

together with the re-estimation of the structure as well as the projection matrices, ensures a Maximum Likelihood estimate, this is not the case when using the approximation introduced in (2.7)

$$\widetilde{\mathcal{C}}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \propto \|\mathcal{P}^\mathsf{T}\mathcal{X} - \hat{\mathcal{Q}}\|^2 + \|\mathcal{P}'^\mathsf{T}\mathcal{X}' - \hat{\mathcal{Q}}'\|^2.$$

By projecting the reprojection error onto the subspace spanned by the columns of $\mathcal{P}$, we obtain an error expressed in 3D space, due to the fact that $\mathcal{Q} = \mathcal{P}^\mathsf{T}\mathcal{X}$ and $\mathcal{Q}' = \mathcal{P}'^\mathsf{T}\mathcal{X}'$. This error is thus similar to (2.16)

$$\mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \propto \|\mathcal{Q} - \hat{\mathcal{Q}}\|^2 + \|\mathcal{Q}' - \hat{\mathcal{Q}}'\|^2.$$

We note however that even though the expressions are identical, the important property that the reconstruction is expressed in an orthonormal basis differentiates the two error functions. As the reconstruction is given up to an arbitrary affine transformation (that is, matrix A in equation (1.21)), the direct 3D error (2.16) depends on the basis in which the structure matrices $\mathcal{Q}$ and $\mathcal{Q}'$ are given. In our case, as the orthogonality of $\mathcal{P}$ and $\mathcal{P}'$ is guaranteed by the orthonormalization process, we can (without approximation) multiply the error by $\mathcal{P}$ and $\mathcal{P}'$ respectively, thus obtaining

$$\begin{aligned}\widetilde{\mathcal{C}}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') &\propto \|\mathcal{P}\mathcal{P}^\mathsf{T}\mathcal{X} - \mathcal{P}\hat{\mathcal{Q}}\|^2 + \|\mathcal{P}'\mathcal{P}'^\mathsf{T}\mathcal{X}' - \mathcal{P}'\hat{\mathcal{Q}}'\|^2 \\ &= \|\check{\mathcal{X}} - \mathcal{P}\hat{\mathcal{Q}}\|^2 + \|\check{\mathcal{X}}' - \mathcal{P}'\hat{\mathcal{Q}}'\|^2,\end{aligned}$$

where $\check{\mathcal{X}}$ and $\check{\mathcal{X}}'$ are the measurement matrices reprojected using the estimated structure and projection matrices from the reconstruction step, that is, rank 3 approximations of $\mathcal{X}$ and $\mathcal{X}'$. The approximated reprojection error has thereby an interpretation with a physical meaning (2D image error expressed in pixels) and differs from the exact expression only in that $\mathcal{X}$ has been replaced by $\check{\mathcal{X}}$.

2.5 Experimental Evaluation

We evaluate our algorithm using simulated and real data. The implementation of all three compared algorithms, that is, FactMLE-EM, TrError and Fact3D, as well as the generation of simulated data, have been done in C++. We also use our approximation to initialize the two nonlinear algorithms TrueMLE and Bundle.

Note that we have chosen not to measure the error with respect to the ground truth 3D affine transformation for several reasons. First, how to measure this error is not obvious, since on the one hand, algebraic measures such as the two-norm of the difference between two affine transformation matrices are not considered meaningful, and on the other hand, measuring distances in the affine 3D space is not possible. Second, we believe that the reprojection error is the right criterion to minimize so as to obtain reliable results. This criterion has indeed been successfully used in many different contexts, and we do not try to investigate its well-foundness.

2.5.1 Simulated Data

We generated m 3D points and two sets of n weak perspective cameras each. The pose of a camera is defined by its three dimensional location, viewing direction and roll angle (rotation angle around the optical axis). The corresponding affine projection matrix is given by a (2×3), truncated, rotation matrix $\bar{\mathsf{R}}_i$ together with a two-dimensional translation vector $\mathbf{t}_i$, both of which premultiplied by an internal calibration matrix. More precisly, we use weak perspective cameras $\mathsf{P}_i = \mathsf{A}_i\bar{\mathsf{R}}_i$ and $\mathbf{t}_i = \mathsf{A}_i\bar{\mathbf{T}}_i$, where A_i is the internal calibration matrix

$$\mathsf{A}_i = k_i \begin{bmatrix} \tau_i & 0 \\ 0 & 1 \end{bmatrix}.$$

The scale factor k_i models the average depth of the object and the focal length of the camera, and τ models the aspect ratio that we choose very close to 1. The 3D points are chosen from a uniform distribution inside a thin rectangular parallelepiped with dimensions $1 \times 1 \times (1 - d)$, and the internal camera scale factors k_i are chosen so that the points are uniformly spread in 400×400 pixel images.

We partition the points into three subsets. The points in the first subset are visible in the first set of cameras only while the points in the second subset are visible in the second set of cameras only. Note that there is no geometrical constraint defining different sectors of the point cloud. The points in the third subset are visible in both camera sets. The third subset contains m_c points, while the two first subsets both contains $m - m_c$ points. Hence, m points are used to perform Structure-from-Motion on each camera set, while m_c points are used for the alignment. The points are projected onto the images where they are visible and gaussian noise with zero mean and standard deviation σ is added.

In order to assess the behavior of the algorithms in the presence of non-perfectly affine cameras, we introduce the factor $0 \leq a \leq 1$. Let Z_{ij} be the depth of the j-th 3D point with respect to camera i, we scale the projected points $\mathbf{x}_{ij}$ by $\mathbf{x}_{ij} \leftarrow \frac{1}{\nu}\mathbf{x}_{ij}$ with $\nu = a + (1 - a)Z_{ij}$, meaning that for $a = 1$, the points does not change and the projection is perfectly affine, and when a tends toward 0, the points undergo stronger and stronger perspective effects. The points are further scaled such that their standard deviation remains invariant, in order to keep them well-spread in the images.

So as to simulate the problem of incomplete data, *e.g.* due to occlusions, we generate a list of missing image points. We introduce the probability p that any given 3D point is occluded in some images and, for simplicity, the same probability that it is occluded in one particular image. This gives a rate of missing data of $\tau = p^2$. We note that since we need at least two views of a 3D point in order to retrieve information about its 3D location, keeping constant the number of points used in the reconstruction, this method of generating the missing points leads to a somewhat lower rate in practice, especially in the case of few cameras.

A 3D model is reconstructed from each of the two camera sets using the factorization algorithm described in Sect. 1.5. Once the camera matrices and 3D points are estimated up to a 3D affine transformation, only the m_c points common to the two camera sets are considered for the alignment of the two reconstructions. We define the overlap ratio of the two camera sets to be $\theta = m_c/m$, that is, for $\theta = 1$ all points are seen in all views, while for $\theta = 0$, the two sets of cameras do not share corresponding points.

Each of the three alignment algorithms yields estimates for the 3D affine transformation and corrected point clouds, except TrError which only gives the transformation. The comparison of the algorithms being based on the reprojection error, the point clouds used to compute it need to be re-estimated so that this error is minimized, given an estimated transformation. This must be done for TrError and Fact3D, but is useless for FactMLE-EM. Our procedure to compute the reprojection error given a 3D transformation is given in appendix 2.3.3.

We setup the following default setting for the simulations: $n = 5$ views, $m = 250$ points, $\theta = 0.2$ (that is, a 20% overlap and $m_c = 50$ points common to the two 3D models), $\sigma = 3.0$ pixels, $d = 0.95$ (flat 3D scene), $a = 1$ (perfectly affine projections) and $\tau = 0.09$. We vary each parameter at a time. Figures 2.5.1, 2.3, 2.4, 2.5, 2.6, 2.7 and 2.8 show the reprojection error averaged over 500 simulations for the algorithms for different parameter values.

In figure 2.5.1, the rate of missing data varies from 0 to 0.5. In order to emphasize the contribution of the EM scheme, we also display the reprojection error of FactMLE-EM after the first iteration. When the rate of missing data grows, the three methods show different tendencies. Whereas FactMLE-EM handles missing data well, the other methods prove to be unstable. However, considering only one iteration of FactMLE-EM, the reprojection error increases just as for the other methods. The difference in performance is thus provided by the EM iterations. In figure 2.3, the overlap ratio varies (coupled with the number of common points m_c, so as to keep the total number of points m constant) from 0.1 to 1.0. The quasi linear behavior should probably be attributed to the increasing size of the model rather than to the increasing overlap.

In figure 2.4 the deviation from the affine model a varies from 0 to 1, away from a perfectly affine projection. Despite the fact that the alignment is affine, even completely projective cameras seem to be well modeled by

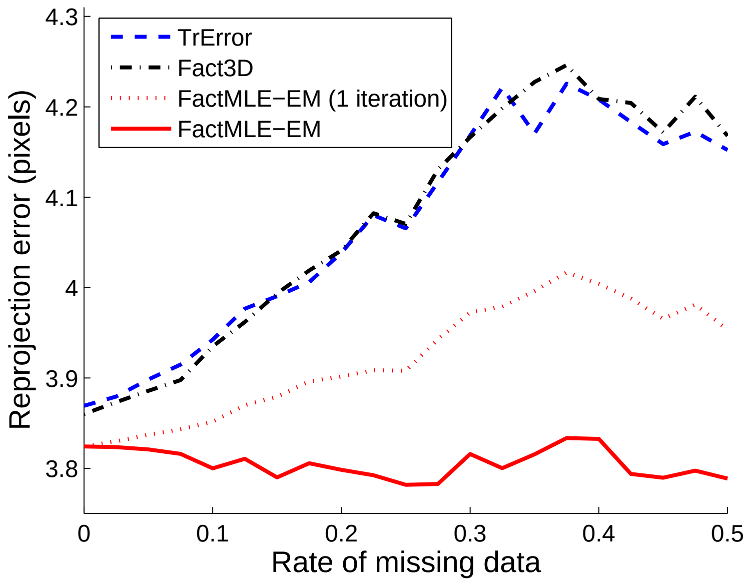


Figure 2.2: Reprojection error against the rate of missing data.

the three methods. In fact, the error induced by the affine approximation is small compared to the added noise in this setting. In figure 2.5 the flatness of the simulated data d varies from 0 to 1, that is, from a cube to a plane. The flatness of the scene does not change the result of the alignment, except for very flat scenes making the algorithms unstable, Fact3D and TrError somewhat more than FactMLE-EM. This result was expected since planar scenes are singular for the computation of a 3D affine transformation.

In figure 2.6, we vary the number of cameras n from 2 to 15. Whereas Fact3D and TrError show almost identical behavior, FactMLE-EM is distinguished by its lower reprojection error. The difference between our method and the other two seems to be more important in the cases where we have few cameras. In figure 2.7, simulations with varying σ reveal a quasi linear relationship between the the noise level and the reprojection error. The slope is somewhat less steep in the case of FactMLE-EM than for the other two methods, indicating that our method is less sensitive to noise.

In figure 2.8, the number of common points m_c varies (coupled with the total number of points m, so as to keep the overlap constant) from 6 to 60. Apart from the results of the methods compared earlier, we also show the result of the minimization using the full SVD decomposition, FullSVD, the true Maximum Likelihood Estimator, TrueMLE, as given by the nonlinear solution of the minimization problem (2.6) and the bundle adjustment over

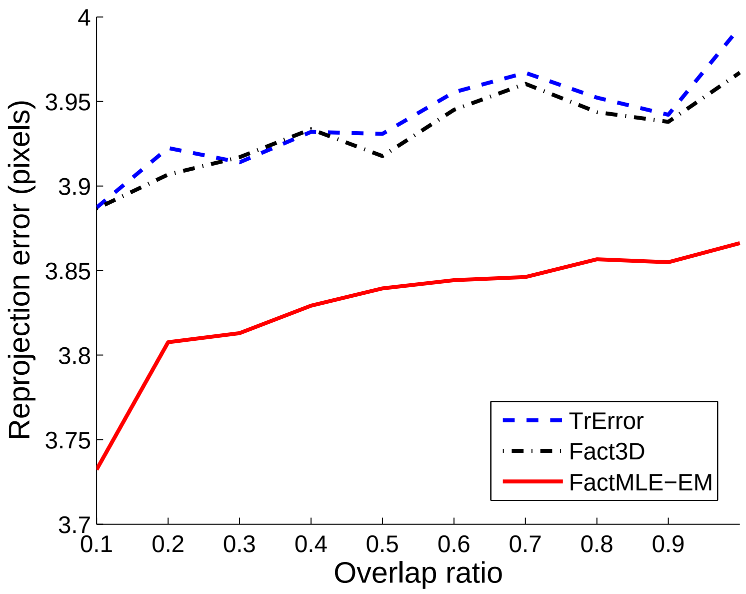


Figure 2.3: Reprojection error against the extent of overlap θ between the two sets of cameras. For $\theta = 1$, all points are seen in all views.the number of points m_c.

the complete set of parameters (see Sect. 2.3.4), Bundle. For the two non-linear methods, we use the results from our algorithm for the initialization. We see that using the orthonormalizing transformations does not impact the results much, since the alternative solution FullSVD needs to perform the minimization in two steps. Moreover, we see that our method yields a good approximation to TrueMLE and a reasonable approximation to Bundle.

Although the three algorithms have similar behavior throughout the sequence of tests, except when varying the rate of missing data, FactMLE-EM consistently outperforms the other ones.

2.5.2 Real Data

We applied the algorithms to real image sequences as follows. A number of images of a scene were taken from different angles and grouped into two sets. Each image plays the role of a camera. A certain number of point correspondences are defined within each one of the image sets, as well as for all the images, thus forming the measurement matrices $\mathcal{X}$ and $\mathcal{X}'$. Hence, not all point correspondences are common to the two camera sets.

The camera is a digital Nikon D100 with a lens of focal length 80−200 mm,

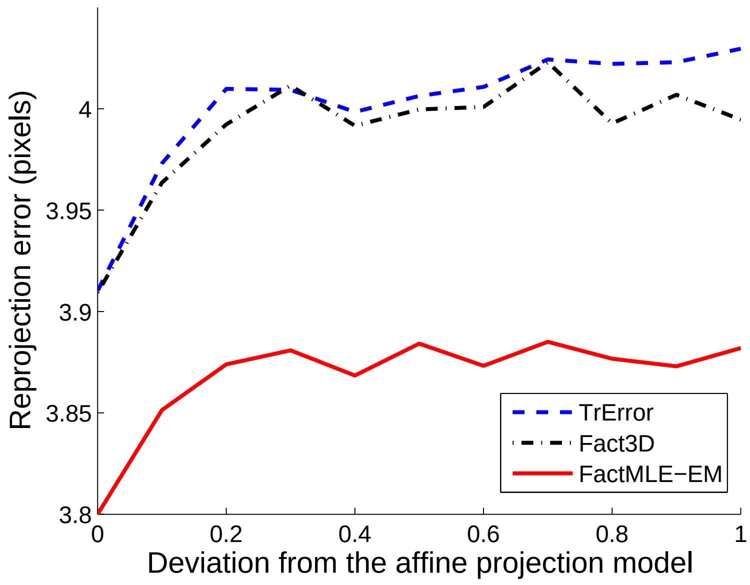


Figure 2.4: Reprojection error against the deviation $1 - a$ from the affine projection model.

giving an image size of 2240×1488 pixels. The camera is not calibrated, that is, the internal parameters are unknown and are furthermore not constant throughout the series of photos, in particular due to the autofocus.

The 'books' sequence. We used a series of images of a rather flat scene, shown on figure 2.9, together with a large set of point correspondences, given by a tracking algorithm, shown on figure 2.9(a). So as to keep the experimental conditions close to the hypothesis of affine cameras, the photos are taken far away from the object using a large zoom (focal length close to 200 mm). This group of images consists of two sets of respectively $n = 2$ and $n' = 3$ images, together with the $m_c = 196$ common point correspondences, and respectively $m = 628$ and $m' = 634$ correspondences for the two sets, giving an approximate overlap of 80%. In addition to the three linear methods, we have also solved the problem using a nonlinear method, in order to present the true Maximum Likelihood estimate. Using the results of our algorithm to initialize the bundle adjustment yields new estimations of the structure and projection matrices and a slightly lower reprojection error. The reprojection errors we obtained are:

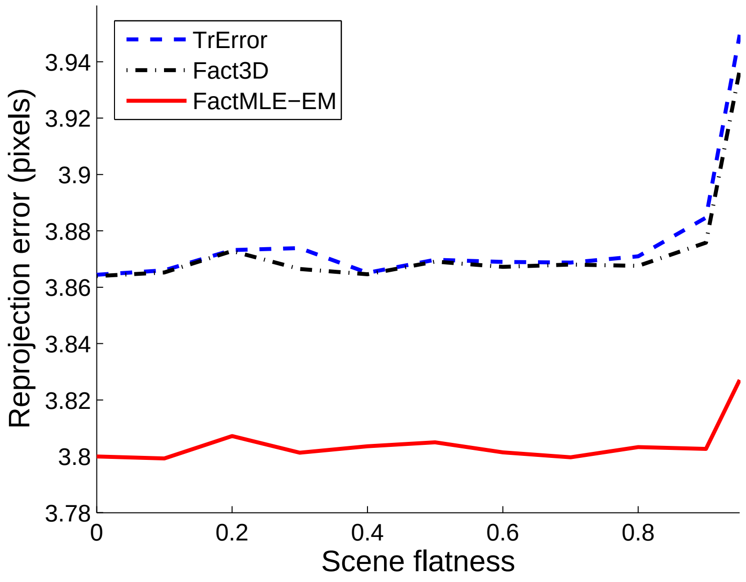


Figure 2.5: Reprojection error against the scene flatness d. For $d = 1$, all 3D points lie on a plane.

TrError	2.17 pixels
Fact3D	1.97 pixels
FactMLE-EM	1.90 pixels
TrueMLE	1.90 pixels
Bundle	1.85 pixels

A detail of an image with the reprojected points due to all three methods is shown in figure 2.9(b). As predicted by the tests on simulated data, FactMLE-EM performs better than Fact3D and TrError. Moreover, we see that FactMLE-EM yields the same reprojection error as the true Maximum Likelihood Estimator and that the result of the bundle adjustment is only slightly better.

The 'cylinder head' sequence. We also tested the algorithms on a series of images of a cylinder head. This sequence was acquired under different conditions than the previous sequence. The photos were taken with the same camera, using a lens with a focal length of 12 mm, at a distance of approximately 60 cm of the object, which is 40 cm long. The points, shown on figure 2.10(b), were manually entered. Using these settings, the affine camera model does not apply and the reconstruction performed prior to the alignment is therefore less reliable. Nevertheless, the result of the alignment is rather

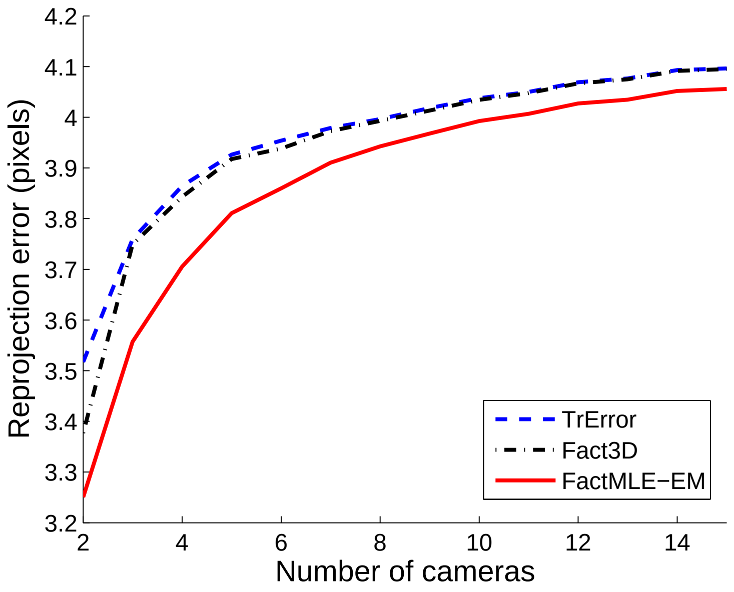


Figure 2.6: Reprojection error against the number of cameras.

good. This group of images consists of two sets of respectively $n = n' = 2$ images, together with the $m_c = 18$ common point correspondences, and respectively $m = 22$ and $m' = 23$ correspondences for the two sets, giving an approximate overlap of 31%. As before, we present the results of the bundle adjustment as well as the true Maximum Likelihood estimate. The reprojection errors were:

TrError	3.78 pixels
Fact3D	3.77 pixels
FactMLE-EM	3.76 pixels
TrueMLE	3.76 pixels
Bundle	3.72 pixels

The two sets of images are displayed in figure 2.10(a) and the given point matches together with the FactMLE-EM reprojections are displayed in figure 2.10(b).

The 'building' sequence. We finally tested the algorithms on a series of images of a building. The point correspondences are again given by a tracking algorithm, but this time the data set is incomplete. We need at least two views of a 3D point in order to use it for the reconstruction, so we keep only those points that are present in two or more images. We then

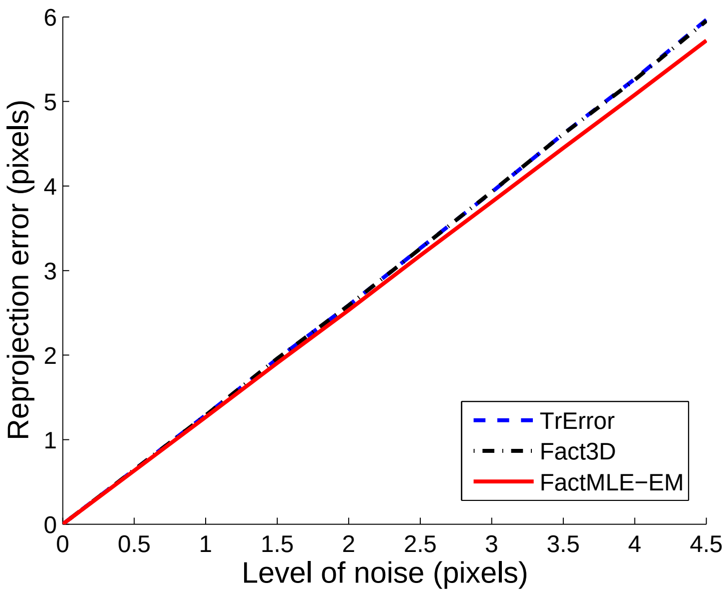


Figure 2.7: Reprojection error against the noise.

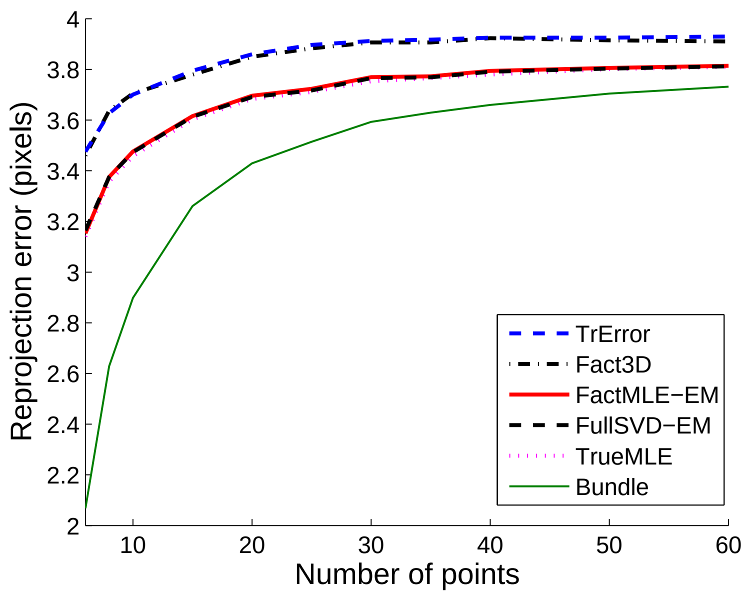
define a point correspondence to be common to the two sets and thus used for the alignment of the two reconstructions, as soon as it is present in (at least two images in each one of) the two sets. This group of images consists of two sets of respectively $n = 5$ and $n' = 5$ images, together with the $m_c = 40$ common point correspondences, and respectively $m = 94$ and $m' = 133$ correspondences for the two sets, giving an approximate overlap of 43% and 30% respectively. The rates of missing data are approximately for the first camera set 31% (13% for the common points) and for the second camera set 22% (11% for the common points). We note that the missing points are essentially not due to occlusions but to failure in the tracking algorithm or to the points being out of range in the images. As before, we present the results of the bundle adjustment as well as the true Maximum Likelihood estimate. The reprojection errors we obtained are:

TrError	0.85 pixels
Fact3D	0.84 pixels
FactMLE-EM	0.78 pixels
TrueMLE	0.77 pixels
Bundle	0.69 pixels

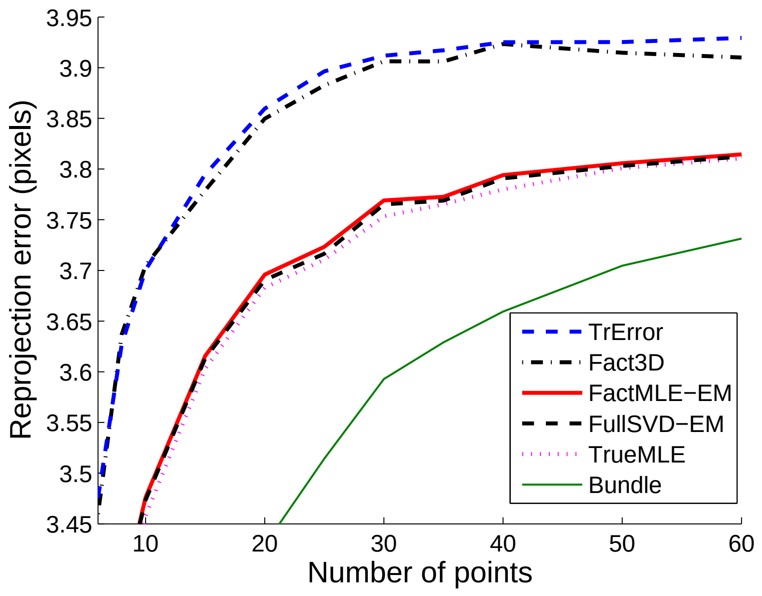
As predicted by the simulations with varying rate of missing data, the difference between the methods is more important when processing incomplete data. Whereas Fact3D and TrError yield similar errors, FactMLE-EM distinguishes itself with a significantly lower error. We note that once more, the approximation of the Maximum Likelihood Estimator is good and that our method is also a reasonable approximation to the bundle adjustment. The results of this sequence are displayed in figure 2.11.

2.6 Conclusions

We presented a method to compute an approximated Maximum Likelihood estimate of 3D affine transformations, under standard hypotheses on the noise distribution, aligning sets of 3D points obtained from uncalibrated affine cameras. The method computes all aligning transformations in a single computation step in the occlusion-free case, by minimizing the reprojection error over all points and all images. An iterative extension is presented for the missing data case. Experimental results on simulated and real data show that the proposed method consistently performs better than other methods based on 3D measurements. Comparison to nonlinear methods shows that the approximation is good. Our algorithm is easy to implement, fast and needs no initialization. It is an essential element for reconstruction methods based on the alignment of partial reconstructions, that is, sequential and hierarchical methods, and can be used to initialize more complex methods.

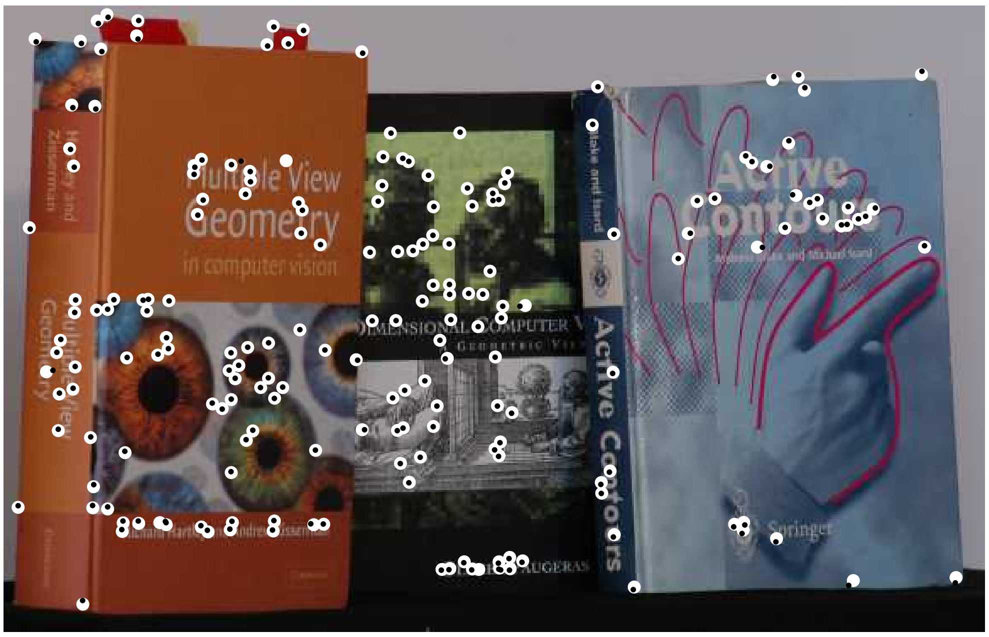


(a)

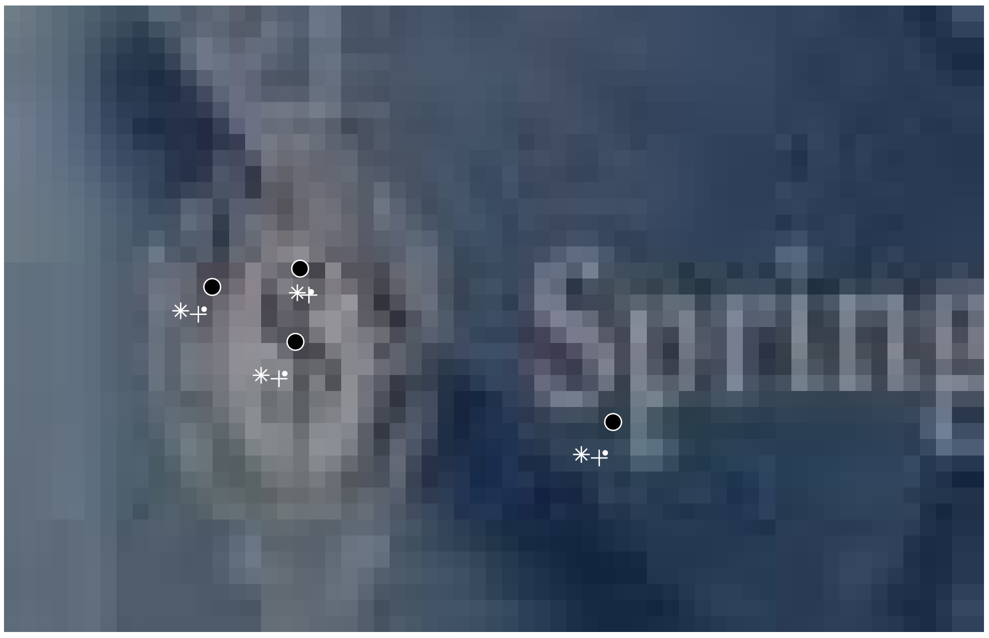


(b)

Figure 2.8: Reprojection error against (a) the number of points. (b) shows a detail.

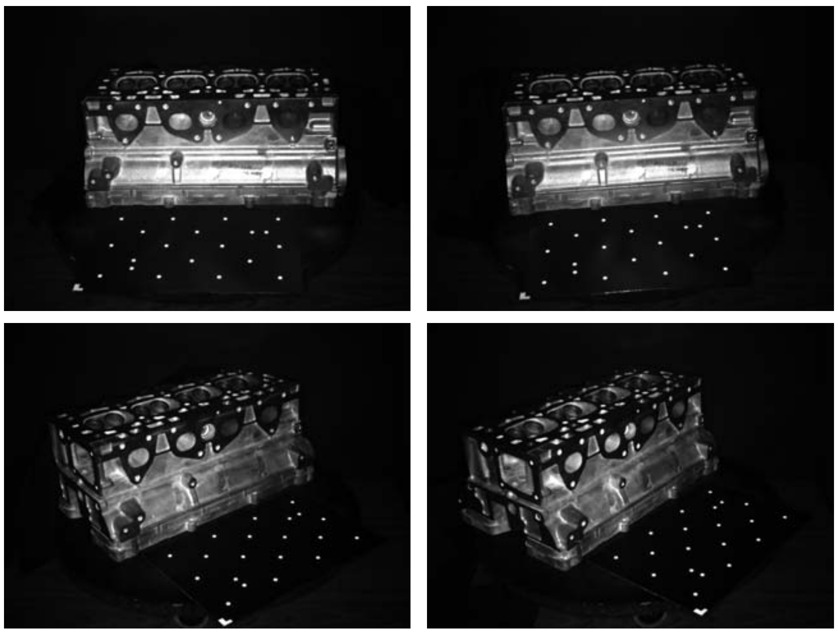


(a) One image from the 'books' sequence overlaid with the $m_c = 196$ point correspondences in white and reprojected points in black.



(b) A detail from the image in (a) showing the original points in black and reprojected points in white, from FACTMLE-EM (points), FACT3D (stars) and TRERROR (crosses).

Figure 2.9: Results from the 'books' sequence.


(a) The two sets of images of the 'cylinder head' sequence.

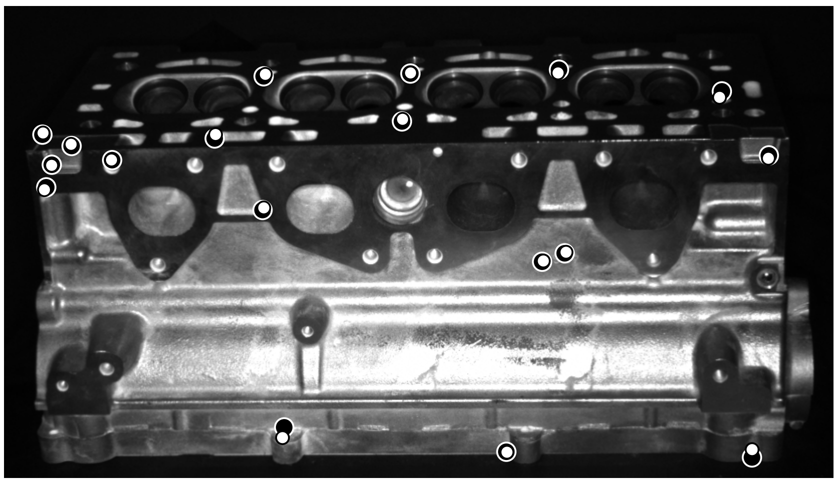
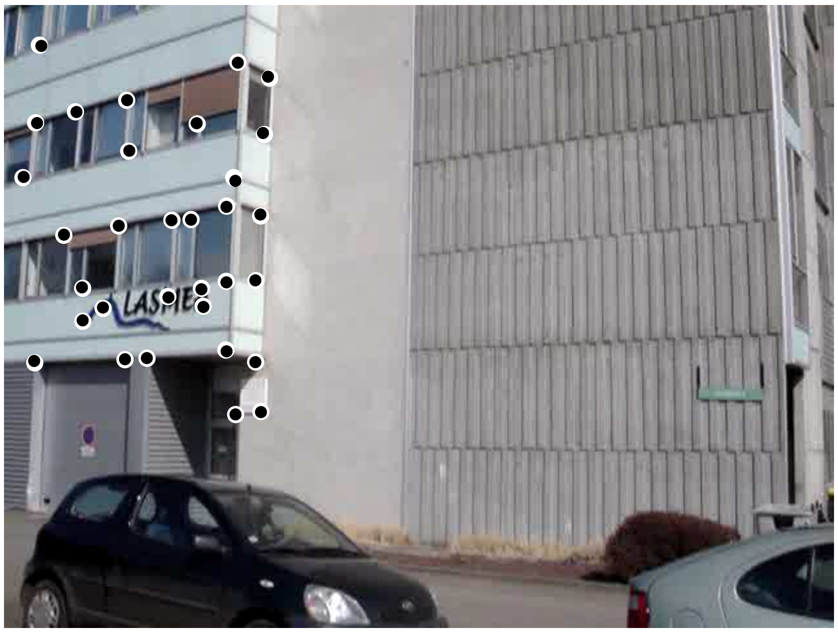
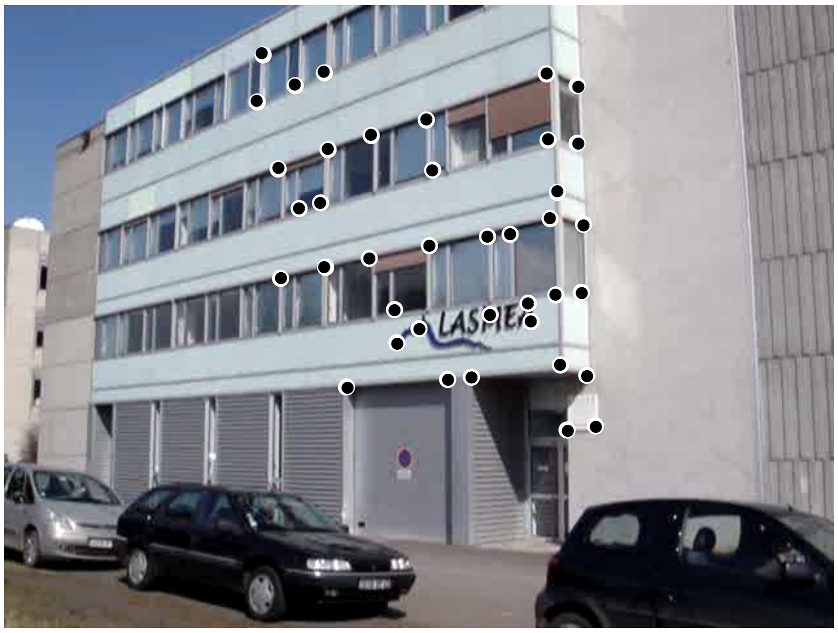

(b) The original points in black together with their reprojections due to FACTMLE-EM in white. The image is a closeup from the first image in the first set.

Figure 2.10: Results from the 'cylinder head' sequence.



(a)



(b)

Figure 2.11: The original common points in white together with their reprojections due to FactMLE-EM in black. The two images are the first ones in the respective camera sets. Note that part of the points used in the alignment are out of range in (a), thus defined as missing.

Chapter 3

Reconstruction of 3D Parametric Curves

In this chapter, we deal with the problem of reconstruction of 3D curves, given the CAD model, for the purpose of controlling conformity with respect to this model. We dispose of a set of images with given perspective projection matrices. The reconstruction will be accomplished by means of the observed contours and their matching, both across the images and to the model.

The parameterization of the curves as well as the optimization algorithms we use must yield an estimate that meets the requirements of accuracy and robustness necessary to perform a control of conformity. We have chosen to use NURBS curves [52], a powerful mathematical tool that is also widely used in industrial applications as well as in the conception chain.

In order to ensure stability, any method used ought to be robust to erroneous data, namely the primitives extracted from the images, since images of metallic objects incorporate numerous false edges due to reflections.

The methods developed have been published in the proceedings of SCIA 2007 [45], EMMCVPR 2007 [43] and WISP 2007 [44] respectively.

Reconstruction de courbes paramétriques 3D

Dans ce chapitre, nous présentons nos travaux sur la reconstruction de courbes paramétriques 3D. Nous présentons un algorithme pour la reconstruction, basée sur le modèle CAO, dans le but de procéder à un contrôle de conformité de l'objet observé par rapport à ce modèle. Nous disposons d'un ensemble d'images calibrées, c'est-à-dire avec les paramètres internes et externes (position et orientation) des caméras connus. La reconstruction sera réalisée par le biais des contours observés dans les images et par rapport au modèle.

La paramétrisation des courbes, ainsi que les algorithmes d'optimisation utilisés, doivent fournir une estimation conforme aux exigences de précision et de robustesse nécessaires pour pouvoir procéder au contrôle de conformité. Nous avons choisi d'utiliser des courbes NURBS [52], outil mathématique puissant qui est également présent dans les applications industrielles ainsi que dans la chaîne de conception des produits manufacturés.

Afin de garantir la stabilité, la méthode doit être robuste aux données erronnées, notamment aux primitives extraites des images, puisque les images des objets métalliques comportent de nombreux contours parasites dus aux réflexions.

Les méthodes développées ont été publiées dans les actes de SCIA 2007 [45], EMMCVPR 2007 [43] et WISP 2007 [44] respectivement.

- Etat de l'art
- Formulation du problème
- Optimisation sur les points de contrôle
  Présentation de notre méthode de reconstruction de courbes 3D pour un nombre de paramètres donné. La reconstruction est faite par le biais d'une minimisation de l'erreur résiduelle dans les images.
- Insertion de points de contrôle
  La méthode de reconstruction pour une paramétrisation fixée est complétée par une procédure d'augmentation de degrés de libertés, par insertion de points de contrôle.
- Optimisation par minimisation d'énergie du gradient
  Présentation d'un autre critère à minimiser et de l'algorithme qui en découle.
- Conclusions

3.1 State of the Art

3.1.1 Active Contours for Stereo Reconstruction

Algorithms based on active contours [36] allow for a local adjustment of the model and a precise reconstruction of primitives. Although initially defined for ordered point clouds, active contours have been adapted to parametric curves in the stereo setting [48]. More precisely, the method allows for an evolution of reprojected model curves toward the image edges, in order to minimize the distance in the images between the predicted curves and the observed edges.

In the area of 3D reconstruction of non-parametric curves, Ghaffari proposes a method based on surface forces and a gradient vector field (GVF [72]) [24]. In an initialization phase of 2D optimization, two independent contours are estimated. This step is followed by epipolar matching and triangulation, thus reconstructing the 3D curve. A comparison between the reprojection of the curve into the images and the original 2D curves yields a displacement vector that will be used to compensate for calibration errors. The 3D active contour finally evolves by means of surface forces, correspondence forces and gradient forces.

In the field of medical imaging, energy minimization methods have been developed to reconstruct 3D curves using a stereo setting. Sbert and Solé reconstruct in [55] a 3D curve using an energy-based evolution method. The associated partial differential equation of the energy functional, derived by the Euler-Lagrange formulation, is solved using a level-set approach. In [9], Canero et al. define a force field by reprojecting external image forces, given by the distance to the edges. A 3D curve is then reconstructed via the evolution of an active contour, guided by the force field.

Cham and Cipolla propose a tracking method based on affine epipolar geometry [10] that reconstructs a parametric curve in a canonical frame using two affine cameras. The result is two coupled snakes. Two models are used, depending on the tracking situation. The first one is a submanifold model, representing all parameters related to the two 2D B-spline curves as a high-dimensional state vector. The evolution of the coupled B-splines, subject to some geometrical criterion, then consist in projecting the state vector onto a submanifold. The second model uses a canonical frame to represent a 'master' curve, which is then projected onto the two image frames, forming the observed 'slave' curves. The evolution of the active contour is now performed in two steps, the deformation of the master active contour, inducing a deformation of the slave active contours, and the modification of the affine transformation that relates the canonical frames to the image frames.

Whereas the latter model is restricted to planar curves, the former allows a more general setting.

In [71], Xiao and Li deal with the problem of the reconstruction of 3D NURBS curves from stereo images. The formulation of the problem features the minimization of the error between the projections of the reconstructed 3D curve and the observed image curves. The optimization is performed on the 3D parameters of the curve. Using the perspective projection of NURBS curves, together with the constraints imposed by the stereo setting, that is, an epipolar constraint and a weight constraint, the problem is reformulated in the image planes. Assuming parallel cameras, stereo rectification techniques can be used to simplify the equations. Furthermore, the assumption of constant weights of the control points of the projected NURBS curves allows one to decompose the problem into three linear least squares problems in the coupled image curve coordinates. The 3D curve is then retrieved via triangulation. Although NURBS curves are used in the problem formulation, the curves are approximated by B-splines for the optimization process. This makes the problem linear, at the expense of loosing projective invariance and thus precision.

Bascle and Deriche presented in 1993 a method for 3D parametric curve reconstruction in a stereo setting [5]. In a first step, a 3D B-spline approximation of a set of given 3D points and line segments is computed by means of a weighted least squares estimation. The complexity of the curve, that is, the number of control points, is chosen so that the least squares error is below some threshold. In order to refine the 3D model, a minimization problem is formulated, using a gradient-based energy defined in the two images. The energy is minimized using Lagrangian dynamics, considering the curve as a massless material, placed in a viscous medium. During the iterative optimization procedure, the 3D positions of the control points are updated by solving a system of partial differential equations, using damping factors modeling the viscosity. These factors are chosen in each step so that the 3D displacement induced correspond to significant 2D displacements of the image curves. The aim is to model only what can be observed from the two available images. This is necessary to avoid instabilities, since the energy minimization is done in the image planes.

In a multi-view setting, Kahl and August suggest in [34] that matching and reconstruction could be coupled. Instead of considering the matching process as a preliminary step, it is interlaced with the reconstruction. So as to take advantage of the coupling, a new curve model is presented, based on an a priori known distribution of the curves and on an image formation model. Given a set of 2D image curves, estimated independently, an inverse problem is formulated aiming at the reconstruction of the 3D curve that gave rise to the image curves. Inspired by the active contours model, the problem is expressed as an energy minimization. The curves are expressed as B-splines, but since the problem is stated independently of the parameterization, other choices are possible. The optimization is done using gradient descent. The algorithm presented allows a completely automatic reconstruction of a set of curves observed simultaneously in multiple images. However, there is no model or structure governing the initial estimation of the images curves and the 3D curves reconstructed are thus just random curves, without correlation.

3.1.2 Reconstruction of Parametric Surfaces

Although we aim to reconstruct 3D curves, other problems related to the reconstruction of parametric structures have come up in the area of surfaces.

In [57], Siddiqui and Sclaroff present a method to reconstruct rational B-spline surfaces. Point correspondences are supposed given. In a first step, B-spline surface patches are estimated in each view, then the surface in 3D, together with the projection matrices, are computed using factorization. Finally, the surface and the projection matrices are refined iteratively by minimizing the 2D residual error. So as to avoid problems due to over-parameterization, the number of control points is limited initially, to be increased later on in a hierarchical process by control point insertion.

3.1.3 Estimation of 2D Parametric Curves

In the case of 2D curve estimation, other aspects of the problem are addressed. Cham and Cipolla adjust a spline curve to fit an image contour [11]. Control points are inserted iteratively using a new method called PERM (potential for energy-reduction maximization). An MDL (minimal description length [27]) strategy is used to define a stopping criterion. In order to update the curve, the actual curve is sampled and a line-search is performed in the image to localize the target shape. The optimization is performed by gradient descent. Brigger et al. present in [8] a B-spline snake method without internal energy, due to the intrinsic regularity of B-spline curves. The optimization is done on the knot points rather than on the control points, which allows the formulation of a system of equations that can be solved by digital filtering. So as to increase numerical stability, the method is embedded in a multi-resolution framework.

In [19], Figueiredo et al. address the problem from a statistical point of view, proposing a completely automatic contour estimator, in the sense that no parameter need to be adjusted by the user. Supposing a uniform distribution of the knot points, the B-spline curve that approximates a given set of contour points at best, in the least squares sense, is given by a linear system depending only on the number of control points. This number is fixed in a last step, where an exhaustive search is carried out over all relevant value and the optimum is chosen using an MDL criterion.

Meegama and Rajapakse introduce in [47] an adaptive procedure for control point insertion and deletion, based on the euclidean distance between consecutive control points and on the curvature of the NURBS curve. Local control is ensured by adjustment of the weights. The control points evolve in each iteration in a small neighborhood (3×3 pixels).

Yang et al. use a distance field computed a priori with the fast marching method in order to adjust a B-spline snake [73]. Control points are added in the segment presenting a large estimation error, due to a degree of freedom insufficient for a good fit of the curve. The procedure is repeated until the error is lower than a fixed threshold. Redundant control points are then removed, as long as the error remains lower than the threshold.

3.1.4 Localization and Tracking

Drummond and Cipolla present, in the framework of a pose estimation algorithm [16], a mono-dimensional edge-tracking method using line-search along the curve normals, initiated at sample points. After a weighting based on the confidence of the edge points, candidate points are used in a linear robust pose estimation. In [66], Vacchetti et al. introduce the use of multiple hypotheses in the image line-search. A new robust estimator is defined for use in an iterative optimization, where the line-search needs to be carried out only once. Stewart deals with the bias problem caused by multiple structures in robust estimation [58]. When outliers have too much coherence to be discarded in an efficient way by classic robust estimators, new estimators must be defined.

3.1.5 Conclusions

Although the problem of 3D reconstruction of curves is addressed in numerous contexts, the application that we aim at still is not fully covered. Based on the existing work, we have therefore developed an algorithm allowing the reconstruction of curves in the industrial context that we target, to the required precision.

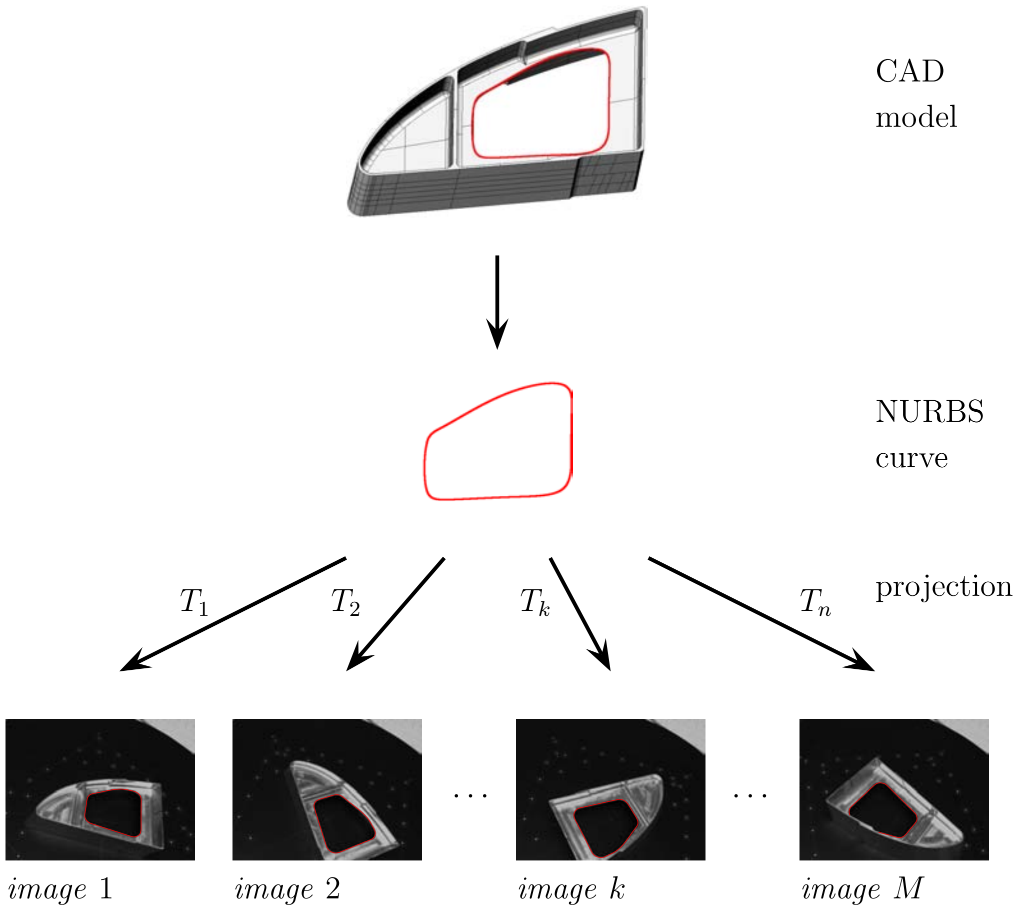


Figure 3.1: The aim of this part of our work is to reconstruct the 3D curves of an industrial object. The target curve is extracted from the CAD model and projected onto the images. The 3D curve is then adjusted so as to fit the observed image contours. It is the 2D information from the images that is used to estimate a 3D curve that is finally compared to the model curve.

3.2 Problem Formulation

Given a set of images of an object, together with its CAD model, our goal is to reconstruct in 3D the curves observed in the images. We suppose that we dispose of the complete camera matrices, giving the rigid transformation, the pose, as well as the intrinsic camera parameters, for each one of the images. The idea is to extract a curve from the CAD model and project it onto the images. Using techniques from image processing, the 3D model curve is then fitted to the observed image curves. The process is depicted in figure 3.1. The aim is to find the 3D curve giving the observed 2D curves. As the NURBS curves provided by the CAD model are suitable for a perspective projection framework, the parametrization is maintained in our method. Another advantage of using parametric curves is the inherent regularity, due to the formulation as piece-wise rational polynomials.

The reconstruction is performed by minimizing a functional, defined based on image data. Regularity aspects can be incorporated as part of the functional, but we will consider that the NURBS curve in itself meets our requirements. The optimization will operate on the control points of the 3D curve, using a goal function that merges the 2D information provided by the set of images.

The reconstructed 3D curve $\mathbf{C}$ is given by the solution to the minimization problem, formulated for a set of M images by

$$\mathbf{C}(\mathcal{P}) = \arg\min_{\mathcal{P}} \sum_{i=0}^{M-1} F(T_i(\mathbf{C}(\mathcal{P}))), \tag{3.1}$$

where F is the image functional to be defined, T_i is the projective operator for image i and $\mathcal{P}$ is the set of control points.

3.3 Optimization of the Control Points

The first step of the optimization process consists in projecting the curve into the images. Since the surface model and the camera matrices are known, we can identify the visible parts of the curve in each image and retain only the sample points corresponding to visible parts. During the iterations, to keep the same cost function, the residual error must be evaluated at the same points at each iteration. Supposing small displacements, we can consider that visible pieces will remain visible throughout the optimization. See figure 3.2 for an example of occlusions due to the 3D structure of the objects.

The optimization of (3.1) is done on the 3D control point coordinates, leaving the remaining parameters of the NURBS curve constant. The weights associated with the control points are modified by the projection giving 2D weights varying with the depth of each control point, according to the formula (1.37), but they are not included in the optimization.

We now need to define the image functional introduced in (3.1). As we work in a calibrated setting, we can measure Euclidean distances and therefore formulate the cost function as a 2D distance from the projected curve to the image contours. An obvious benefit of this is the direct physical interpretation thus given to the value of the cost function. Since curve to curve distance is a somewhat vague concept, we will rather consider a sampling of the projected curve and point-wise distances to image contours.

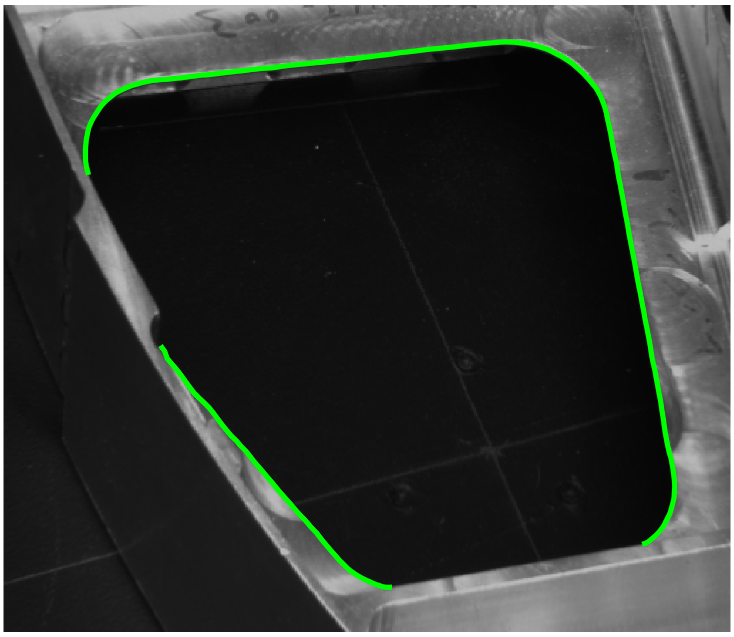


Figure 3.2: Due to the thickness of the object, auto-occlusions cause parts of the curve to be invisible from some viewpoints.

The formulation of the curve estimation as a distance minimization induces a need for a method to identify a set of contour points associated to the projected curve.

3.3.1 Distance Minimization

Using a distance formulation and the properties of NURBS curves, the minimization problem (3.1) is written

$$\min_{\{\hat{\mathbf{P}}_l\}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left\| \mathbf{q}_{ij} - \sum_{l=0}^{n} T_i(\hat{\mathbf{P}}_l) R_{l,k}^{(i)}(t_j) \right\|^2 , \tag{3.2}$$

where $\mathbf{q}_{ij}$ is a contour point associated with the curve point of parameter t_j in image i, $R_{l,k}^{(i)}$ are the basis functions for the projected NURBS curve in image i and T_i is the projection operator for image i.

Although the context is not the same, we recognize the minimization formulation from the problem of Structure from Motion, see 1.5. Indeed, the metric bundle adjustment consists of minimizing a sum of image distances, where the sum is taken over all images and all data points. If the image error is zero-mean and Gaussian then bundle adjustment is the Maximum

Likelihood Estimator. In our case, for the image error to satisfy this, the method used to identify the image contour points must yield a non-biased estimation. In the presence of other contours in the neighborhood of the one corresponding to the current curve, this is difficult to achieve.

Each contour point $\mathbf{q}_{ij}$ is associated with a curve point and thus depends on the current value of the control points. The minimization problem given is therefore nonlinear and cannot be solved by the linear least squares method. An iterative method, such as the Levenberg-Marquardt method, must be used. Note that a new set of matching contour points must be computed at each iteration. A common feature among implementations of the Levenberg-Marquardt method is to allow the user to provide the value of the jacobian matrix in the current point, in addition to the function value. This is meant to prevent the approximation error and the additional computations that a finite difference approximation implies, if a closed-form expression exist. In the case of NURBS curves, the nature of the basis functions as being rational polynomials ensures that the first-order derivatives of the curve can be computed exactly.

3.3.2 The Search for Image Contours

There are several image processing methods available to retrieve the contours of an image, most of them being based on the gradient image. As we will need to carry out the search repeatedly, we want to find a quick way to do it, but without compromising the precision. Due to the nature of our applications, we can suppose that the possible deformations that we are looking for are small. This, together with the fact that we have an approximate location of the curve, allows us to settle for performing a local search around the projected curve.

The search for candidate contour points is carried out independently in the images using a method inspired by the one used by Drummond and Cipolla in [16]. We sample the NURBS curve projected in the image, to use as starting points in the search for matching contour points. A line-search is performed in order to find the new position of the curve, ideally corresponding to an edge. Our approach is based solely on the contours. Due to the aperture problem, the component of motion of an edge, tangent to itself, is not detectable locally and we therefore restrict the search for the new edge position to the edge normal at each sample point. The pointwise search is illustrated in figure 3.3. As we expect the motion to be small, we define a search range (typically in the order of $5-10$ pixels) so as to limit computational cost. In order to find the new position of a sample point, for each point belonging to the normal within the range, we evaluate the

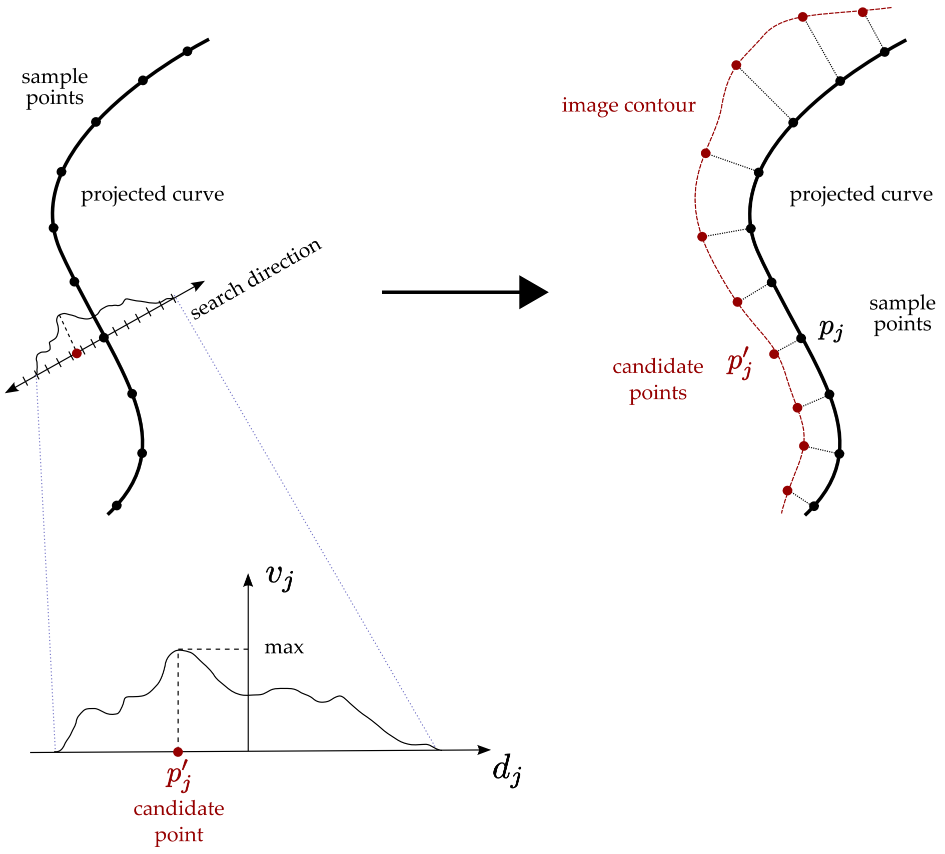


Figure 3.3: The search for candidate points is carried out pointwise using line-search in the direction of the curve normal.

gradient and compute a weight based on the intensity and the orientation of the gradient and the distance from the sample point. The weight function v_j for a sample point $\mathbf{p}_j$ and the candidate point $\mathbf{p}_\xi$ will be of the form

$$v_j(\mathbf{p}_\xi) = \varphi_1(\|\nabla \mathbf{I}_\xi\|) \cdot \varphi_2\left(\frac{\hat{\mathbf{n}}_j \cdot \nabla \mathbf{I}_\xi}{\|\nabla \mathbf{I}_\xi\|}\right) \cdot \varphi_3(\|\mathbf{p}_j - \mathbf{p}_\xi\|),$$

where $\hat{\mathbf{n}}_j$ is the normal of the projected curve at sample point j, $\nabla \mathbf{I}_\xi$ is the gradient at the candidate point and the φ_k are functions to define. The weight function will be evaluated for each candidate $\mathbf{p}_\xi$ and the point $\mathbf{p}'_j$ with the highest weight, identified by its distance from the original point $d_j = \|\mathbf{p}_j - \mathbf{p}'_j\|$, will be retained as the candidate for the new position of the point.

The bounded search range and the weighting of the point based on their distance from the curve yield a robust behavior, close to that of an M-estimator.

The search along the normal to the curve will necessarily be performed in discreet steps. However, if the true optimal position is located between two points, in addition to the obvious loss in precision, we may also face an oscillation problem. For these reasons, we need to consider the neighborhood of the retained point. We will use a quadratic interpolation of the curve based on the measured values, that is, we fit a second degree curve around the maximum and retain the position of the maximum of this curve, see figure 3.4. The aim is to reach a sub-pixel precision in the images.

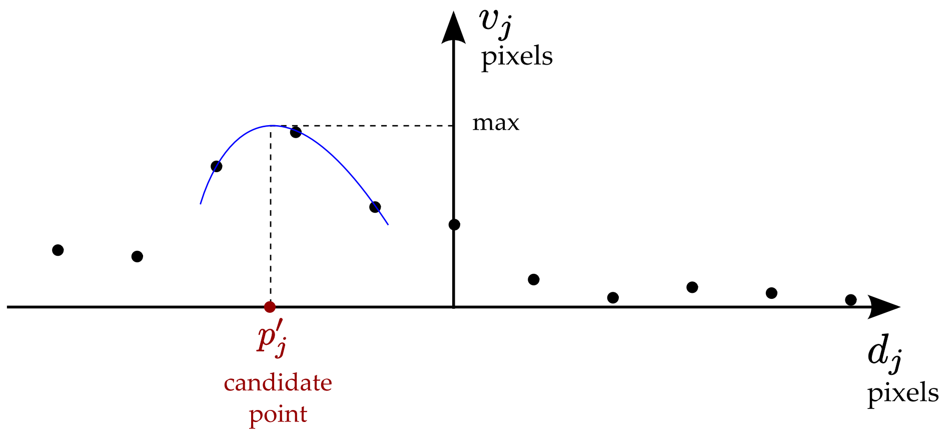


Figure 3.4: A second degree curve is fitted around the point with the highest value, using the values of the neighboring points along the line. The maximum of this curve is chosen as the optimal displacement.

3.3.3 Conclusions

The curve reconstruction method we presented yields a 3D NURBS curve whose projections coincides, at least approximately, with the observations in the images. The approximation comes from the fact that the parameterization induced by the curve extracted from the model, namely the number of degrees of freedom, may be insufficient to describe the observed real world curve. Indeed, an imperfection in an industrial object would most certainly be rendered as a rough portion of an elsewhere smooth curve. In order to overcome this problem, we want to increase the number of degrees of freedom. This must however be done with caution, since too many parameters would yield an over-fitting of the model.

3.4 Control Point Insertion

The problem has two parts. First, the optimization of the 3D NURBS curve by solving a minimization problem on a fixed number of control points, then the control point insertion procedure. For the fixed size optimization problem, we proceed as before. This step allows the control points to move in 3D, but does not change their number. In order to obtain an optimal reconstruction of the observed curve, we iteratively perform control point insertion. So as to avoid over-parameterization for stability reasons, the first optimization is carried out on a limited number of control points. Their number is then increased by iterative insertion, so that the estimated 3D curve fits correctly also in high curvature regions. Due to the characteristics of NURBS curves, the insertion of a control point is done without influence on the curve and each insertion must therefore be followed by an optimization in order for the curve to take advantage from the increased number of degrees of freedom.

Due to the use of NURBS curves, we have a method to insert control points, see section 1.6.5 for details. What remains is to decide where to place them. We also need a criterion to decide when to stop the control point insertion procedure.

3.4.1 Position of the New Control Point

The parametrization of NURBS curves is based on a knot vector, splitting the range of the free parameter into intervals. The influence of each control point being closely linked to these intervals, we will refer to them in the insertion process. When deciding where to place the new control point, we will thus relate to the intervals of the curve.

In the literature, a common approach is to consider the dual problem of knot point insertion. As the knot point itself has no representation in the image, the location on the curve corresponding to the knot point, called the hinge point, will be used. Lu and Milios propose in [39] to insert the control point in such a way that a new hinge is formed at the point on the curve that presents the largest displacement from the data. This will certainly allow for a decreasing residual error around the inserted control point, but the behavior can be unstable for several reasons. Namely, if the chosen point on the curve is already a hinge (or close to one), the insertion of another hinge will change the regularity aspects of the curve. Indeed, at a multiple knot point of multiplicity m, the curve is only $k - m$ times continuously differentiable, where k is the degree. Moreover, the maximum of the curve-data distances, computed pointwise, gives a poor indication of where the model complexity is insufficient. Dierckx suggests in [15] to place the new point at the interval that presents the highest mean error. This is consistent with an interpretation of the error as the result of a lack of degrees of freedom that inhibits a good description of the curve. If, however, the error derives from other sources, this solution is not always optimal. For instance, a significant mean error could also indicate the presence of parasite edges or that of a parallel structure close to the target curve. In [11], Cham and Cipolla define an error energy reduction potential and propose to place the knot point so as to maximize this potential. As the computational cost of this potential is small compared to the cost of the active contour optimization used to find the location of the control points, an exhaustive search is performed over all sample points. The sample point that presents the largest energy reduction potential is retained for the insertion of a hinge.

In our algorithm, since every insertion is followed by an optimization that adjusts the control points, we settle for choosing the interval where to place the point. Since the exact location within the interval is not critical, the point is placed at its midpoint. We will choose the interval with the highest median error, over all images. The error is defined as the distance from a sample point to its corresponding contour point in the image. The search for candidate contour points is carried out using the method described in section 3.3.1.

3.4.2 Controlling the Complexity

One of the motives for introducing parametric curves was to avoid treating all curve points, as only the control points are modified during the optimization. If the number of control points is close to the number of samples, the benefit is limited. Too many control points could also cause numerical instabilities, due to an over-parameterization of the curve on the one hand and the size of the nonlinear minimization problem on the other hand. It is thus necessary to define a criterion that decides when to stop the control point insertion. A short review of the different techniques used in the area of model complexity selection using a statistical approach is given in section 1.7.2.

In the iterative control point insertion procedure of Cham and Cipolla [11], the stopping criterion is defined by means of MDL. In a Bayesian framework, using a MAP estimation assuming normally distributed errors, an expression for the ideal code-length to describe the data and the model is obtained. The criterion depends, on the one hand on the number of control points and on the residual errors, on the other hand on the number of samples and on the covariance. If unavailable, the covariance matrix is replaced by an unbiased approximation of the uniform variance, multiplied with an identity matrix. Iteration is continued until the code-length increases. In order to avoid local minima, the minimum is supposed found once the value remains higher than the last three iterations.

Another way of implementing the MDL method is given by Figueiredo et al. in [19]. The curve fitting is addressed using a multi-layer hierarchical scheme of nested algorithms, where the inner schemes solves different optimization problems for a fixed number of control points. The solution with respect to the number of control points is found through an exhaustive search over all values in a predefined range. A comparison of the estimated curves is carried out using an MDL criterion derived from a Bayesian viewpoint. It is defined by the residual error, the number of samples and the image size, where the image size is considered to be a scale measure.

In the area of parameter estimation for 3D reconstruction, Viéville et al. have used the AIC criterion for the complexity control [67, 68].

Kanatani derives in [35] a criterion called the geometric AIC, meant to serve in the process of degeneracy detection. By evaluating the predictive capability of an estimated model, in terms of the expected residual and the geometric AIC, singularities causing the model to be unsuitable can be detected. In the context of Structure-from-Motion, the singularities can typically be coplanar or colinear points.

Within the framework of epipolar geometry, Torr presents a method for approximating the posterior probability of a model [63]. The model selection paradigm developed, similar to the penalized likelihoods used in the aforementioned methods, is particularly well adapted to situations involving large numbers of latent variables.

As our curve estimation algorithm is computationally expensive and furthermore embedded in an iterative control point insertion process, a stopping criterion based on cross-validation is unfeasible. Our choice of method is therefore restricted to the criterion-based ones. We have chosen to evaluate several possibilities, namely BIC, AIC and the 2D image error. The different criteria will be computed using the contour points found with the method presented in section 3.3.1. The choice of BIC is justified by the asymptotic characteristics, that assures convergence to the model that generated the data as the amount of data tends to infinity. The expression for BIC in the case of normally distributed errors

$$BIC = k \ln n + n \ln \frac{RSS}{n}, \tag{3.3}$$

where k is the number of control points, n is the number of data points and RSS is the sum of the squared residual errors. The use of AIC is supported by a different asymptotical behavior, that is, the ability to choose the model that has the best likelihood for future data. In our setting, the AIC can be written

$$AIC = 2k + n \ln \frac{RSS}{n}. \tag{3.4}$$

Using the 2D image error consist in considering only the evolution of the measurable error throughout the iterations, not explicitly taking into account the numbers of parameters of the model. We consider the rms error, which is given by

$$rms = \sqrt{\frac{RSS}{n}}. \tag{3.5}$$

The three criteria retained will be tested and compared in Sec. 4.

3.4.3 Sampling of the Curve

Each new control point adds an interval to the curve. In order to increase the resolution in regions where the curve undergoes changes rather than globally, the sampling follows the intervals. The insertion of a control point also increases the number of sample points. Furthermore, the sampling is adapted to the visibility of the interval in the images. First, the visibility of each sample point is verified, so that any point occluded by the object itself is removed, see section 1.8. Moreover, the number of samples is based on the projected length of the interval and not on the length of the interval on the 3D curve. Each 2D curve is sampled separately per interval, with a lower limit on the mean distance between sample points of one pixel. This is to avoid giving too much weight to barely visible intervals. Adding a higher limit on the mean distance between sample points reduces the effect of the adaptation following the insertion of control points, but adding more sample points at an earlier stage of the procedure ensures a faster convergence. A tight interval for the mean distance per interval yields a close to uniform image-based sampling.

3.4.4 Algorithm

The algorithm we implemented has two layers. The optimization of a curve using a fixed complexity model is embedded in an iterative structure that aims to increase the number of control points. The nonlinear optimization of the 3D curve is performed by the Levenberg-Marquardt algorithm, using a cost function based on a distance formulation as in section 3.3. The control point insertion procedure uses a search for contour points in the images in order to compute the median as well as the RSS error of the projected curve. The mechanism of our method is outlined in Tab. 3.1.

3.4.5 Conclusions

We have now presented a complete curve reconstruction algorithm, that is able to choose an appropriate model complexity to represent the observed curve. The method is formulated using a distance-based cost function, giving the residual error a physical meaning. However, the distance-based approach is not the only existing solution to the problem of curve estimation. Another method is to define an energy potential in the image and search the curve that minimizes the energy. In order to compare the two strategies and, hopefully, find an even closer match to the observed curve, we will now extend our method to include an alternate cost function based on an energy formulation.

3.5 Maximizing the Image Gradient Magnitude

In the literature, there are two classical methods for the estimation of curves from image data, the distance formulation and the energy formulation. The energy formulation is the base of the classic active contours algorithm, introduced by Kass et al. in 1987 in the seminal article [36]. It is widely used in the medical imaging sector, due to the inherent blurry aspect of the images, making precise edge detection hard. Indeed, the contours that are to be detected are often noisy and present low contrast compared to the more marked edges in images from an industrial context. In this section, we will define an energy potential that will be the cornerstone of an alternate cost function, used in the fixed complexity reconstruction algorithm. The aim is to compare the two methods, in order to choose the one that is the most appropriate for the task.

3.5.1 Definition of an Energy Functional

The classical energy formulation consists in constructing an energy potential defined in each pixel of the image. The energy of a curve is given by the integral of the potential along the curve. The ideal location is the one that minimizes the integral, which is replaced by a sum over all sample points. We obtain a formulation similar to the previous one. Using the properties of NURBS curves, the minimization problem (3.1) is written

$$\min_{\{\hat{\mathbf{P}}_l\}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} E\left(\sum_{l=0}^{n} T_i(\hat{\mathbf{P}}_l) R_{l,k}^{(i)}(t_j) \right), \tag{3.6}$$

- **Visibility Check** Identification of the visible parts

$$\chi_{ij} = \begin{cases} 1 & \text{if } \sum_{l=0}^{n-1} T_i(\mathbf{P}_l)\mathbf{R}_{l,k}^{(i)}(t_j) \text{ visible} \\ 0 & \text{else} \end{cases}$$

- **Line-search** for contour points $\mathbf{q}_{ij}$ matching $\mathbf{p}_{ij} = \sum_{l=0}^{n-1} T_i(\mathbf{P}_l)\mathbf{R}_{l,k}^{(i)}(t_j)$

$$\mathbf{q}_{ij} = \arg_{\mathbf{p}_{ij}^m} \max_{-d \le m \le d} v_j(\mathbf{p}_{ij}^m,) \text{ where } \mathbf{p}_{ij}^m = \mathbf{p}_{ij} + m \cdot \hat{\mathbf{n}}_{ij}$$

- **Optimization** of the control points

$$\min_{\{\hat{\mathbf{P}}_l\}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \chi_{ij} \left\| \mathbf{q}_{ij} - \sum_{l=0}^{n-1} T_i(\hat{\mathbf{P}}_l)\mathbf{R}_{l,k}^{(i)}(t_j) \right\|^2$$

- **Computation of** *BIC*

$$BIC_0 = k \ln(N \cdot M) + N \cdot M \cdot \ln \left(\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \chi_{ij} \left\| \mathbf{q}_{ij} - \sum_{l=0}^{n-1} T_i(\mathbf{P}'_l)\mathbf{R}_{l,k}^{(i)}(t_j) \right\|^2 /(N \cdot M) \right)$$

- **do** (control point insertion)
  - **Line-search** for contour points $\mathbf{q}_{ij}$
  - **Computation of the median error** for each interval I_K.

    $$m_K = \operatorname*{med}_{E_K} |\mathbf{q}_{ij} - \sum_{l=0}^{n-1} T_i(\mathbf{P}'_l)\mathbf{R}_{l,k}^{(i)}(t_j)|$$

    where $E_K = \{(i,j) \,|\, 0 \le i < M, t_j \in I_K, \chi_{ij} \neq 0\}$.
  - **Knot point insertion** at the midpoint of interval $I = \arg\min_K m_K$.
  - **Visibility Check** Identification of the visible parts
  - **Optimization** on the control points
  - **Computation of** BIC_J
- **while** $(BIC_J < BIC_{J-1})$

Table 3.1: The reconstruction algorithm presented.

where $R_{l,k}^{(i)}$ are the basis functions for the projected NURBS curve in the i^{th} image.

In the active contours context, the energy is often split into two parts, an internal and an external one. The internal energy restricts the curve in terms of continuity and smoothness, whereas the external part models the attraction toward the image contours. When using NURBS curves, the energy functional E can be restricted to its external part, due to the regularity aspects incorporated in the parametrization. Excluding special cases, namely the presence of multiple knot points, the continuity is assured and keeping the curvature at reasonable levels is an issue of the relative location of the control points rather than of the energy functional. Concerning the external energy, a common choice is to use the gradient intensity. We will however include local information on the curve as well, namely its normal direction, using the intensity of the gradient projected onto the curve normal. This choice is motivated by the aperture problem, according more importance to the normal component than to the tangential component of the image gradient.

3.5.2 Distance versus Gradient Energy

For comparison, we have implemented the two methods in the iterative setting. Both methods yielded similar results and converged after a number of iterations to an asymptotic lower limit. The 3D error with respect to the true curve is however somewhat lower for the gradient-based method. The results are given in figure 3.6. The difference is partly explained by the noise and the parallel structures perturbing the edge tracking algorithm. An example of candidate points located on a parallel image contour, due to specularities, is given in figure 3.5. More details on the tests comparing the two cost functions are given in chapter 4. Although the gradient intensity method outperforms the distance method, the distance-based cost function will prove to be useful in the iterative framework that will embed the curve optimization.

3.5.3 Hybrid Optimization

Whereas the energy approach allows us to obtain a more precise reconstruction of the curve, the distance method allows a faster convergence from a bad initialization. In order for the energy-based cost function to attract the model curve to a far-off target, the gradient must be smoothened out, which impairs the precision as well as the convergence rate. In order to take advantage of the strength of each one of the methods, we have chosen to use a hybrid algorithm, combining the two. In a first step, an initial minimization

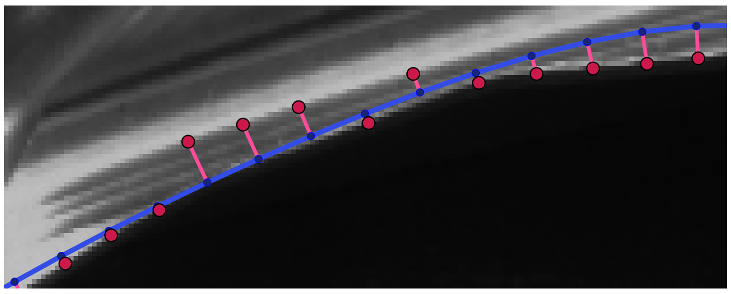


Figure 3.5: Problems related to specularities and to the search for candidate points. Starting at the projection of the initial curve (in blue), some candidate points (in magenta) belong to a parasite edge.

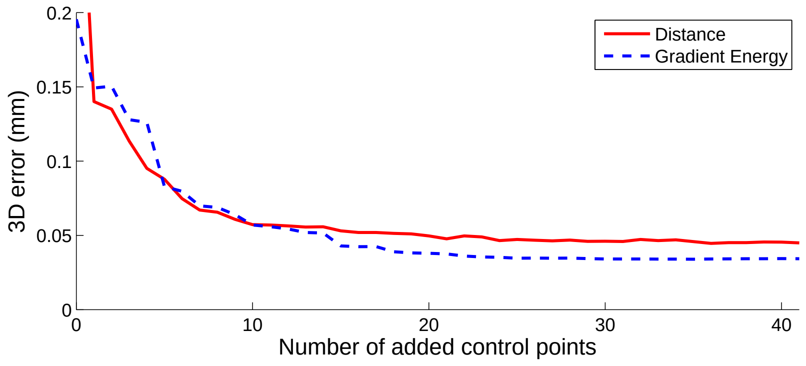


Figure 3.6: The evolution of the error, with respect to the true 3D curve, for an optimization using cost functions based on the distance to the image contours and on the gradient intensity respectively. Whereas the distance method seems to yield good results in the start, its asymptotic limit is somewhat higher than that of the gradient intensity method.

is performed using the distance-based cost function. Then, during the control point insertion, the energy-based cost function is used.

So as to settle the hybrid approach and in order to tune the parameters, different strategies will be tested and compared in Chap. 4. An overview of the hybrid algorithm with its final formulation will be given in the end of the evaluation.

3.6 Conclusions

We have presented an adaptive 3D reconstruction method using parametric curves, limiting the degrees of freedom of the problem. An algorithm for

3D reconstruction of curves using a fixed complexity model is embedded in an iterative framework, allowing an enhanced approximation by control point insertion. The optimization of the curve with respect to the control points is performed by means of a minimization of an gradient-based energy functional, whereas the insertion procedure is based on the distance from the curve to the observed image contours.

Chapter 4

Experiments on 3D Curve Reconstruction

We present in this chapter the results of the evaluation of our method for the reconstruction of parametric curves. In order to test our algorithms and quantify the performances, we use virtual as well as real images. Whereas simplified virtual objects serve to estimate the theoretical limits of our algorithms, images of complex manufactured objects presenting the difficulties targeted by our approach allow us to evaluate the performance in an industrial context. We finally illustrate the behavior of our system in the presence of anomalies, using simulated deformations of the target object. Using a series of virtual images of an object presenting a minor anomaly, we evaluate the capacity of our system to detect nonconformities.

Evaluation expérimentale

Nous présentons dans ce chapitre les résultats de l'évaluation de notre méthode sur la reconstruction de courbes paramétriques pour des objets métalliques complexes. Afin de valider notre approche et de quantifier les performances, nous utilisons des données simulées et des images réelles. Tandis que des objets virtuels simplifiés servent à estimer les limites théoriques de nos algorithmes, des images d'objets manufacturés complexes présentant les difficultés visés par notre approche nous permettent d'évaluer la performance de la méthode dans le contexte industriel. Enfin, nous avons simulé des anomalies sur des objets afin de valider la capacité de notre système à détecter des non conformités.

- Préliminaires
  Présentation du procédé de génération des images virtuelles, des erreurs mesurées et des paramètres.
- Images Virtuelles
  Essais sur des données de synthèse afin de valider notre méthode et de choisir la stratégie et les paramètres de l'algorithme.
- Images réelles
  Essais sur des images d'objets manufacturés complexes afin d'évaluer la performance de notre algorithme dans un contexte industriel.
- Nonconformités
  Essais sur des images d'un objet virtuel présentant des anomalies.
- Algorithme
  Récapitulatif des choix effectués et présentation de l'algorithme final.
- Conclusions

4.1 Preliminaries

4.1.1 Generation of Virtual Images

In order to validate the algorithms used for image data extraction and for curve reconstruction, we perform a number of tests on virtual images. The virtual setting will also be useful to simulate deformations of the target object. The use of images of deformed virtual objects enlarges the possibilities for testing. To this end we have implemented a procedure that allows us to visualize a CAD model, generating a series of virtual images, together with the camera parameters for each view.

4.1.2 Curve to Curve Error Measures

So as to perform a quantitative evaluation of our method, we need to be able to compute some kind of error. In the case of the 3D reconstruction of a single curve, this error should be a measure of the difference between the real curve and the estimated one. For testing purposes, we will use a setting where the ground truth is known. At the end of our reconstruction procedure, we thus have two 3D curves that we wish to compare. Since curve to curve distances is a mathematically vague concept, we consider a sampling of the estimated curve and point-wise distances to the target parametric curve.

Our conception of the curve to curve distance yield a high-dimensional vector of errors after completion of the optimization process. For the evaluation, aiming at a good description of the error, we present it on several forms. For an error vector $\mathbf{e} = [e_0 \cdots e_n]^t$, where n is the number of sample points of the 3D curve, we have

- the **rms (root mean square) error**

$$rms = \sqrt{\frac{\sum_i e_i^2}{n}}$$

- the **mean error**

$$\bar{e} = \frac{\sum_i e_i}{n}$$

- the (sample) **standard deviation**

$$s = \sqrt{\frac{\sum_i (e_i - \bar{e})^2}{n-1}}$$

The distance between the model curve and the reconstructed one is something that we wish to evaluate after the reconstruction is completed. However, during the computations, some other measure is necessary in order to decide whether to process has converged or not. This measure will be based on the 2D image distances from the projections of the estimated curves to the observed image curve points. The curve points are given by the line search algorithm described in Sec. 3.3.2. In the results section, we give the 2D rms error, expressed in pixels, considering the projections in all images. The error vector thus contains all visible sample points in all images, without weighting. The expression for the rms error is the same as in the 3D case.

4.1.3 Parameters

There are a certain number of parameters of the algorithm that need to be set. Some of them are related to the image and the size of the object in pixels, such as the value of the smoothing factor in the Deriche gradient filter or the size of the search range in the contour detection. Others are related to the shape of the curve, such as the maximum number of control points to be inserted or the insertion strategy. These parameters will be set independently, depending on the experimental settings. It is expected that these parameters remain stable for the same kind of applications.

4.2 Virtual Images

Setting

We construct a simplified model of an object, based on a single target curve inserted in a plane. Using a virtual camera with known parameters, internal as well as external, the object is visualized from different viewpoints. The set of virtual images thus created is shown in figure 4.1. The image size is 1284×1002 pixels. For the computations, we use the complete series of 21 images, but also subsets of varying sizes in order to confirm the results obtained.

The reconstruction algorithm is initialized using a modified model curve, parameterized by 10 control points, shown in figure 4.2 together with the model curve. The geometry of the model curve indicates that 10 control points is not enough to correctly describe it. The initial sampling used for the computations is of 200 points. To fix the scale, note that at the mean distance from the object curve, one pixel corresponds roughly to 0.22 mm on the object in a fronto-parallel view and that the largest dimension of the curve is of 118 mm.

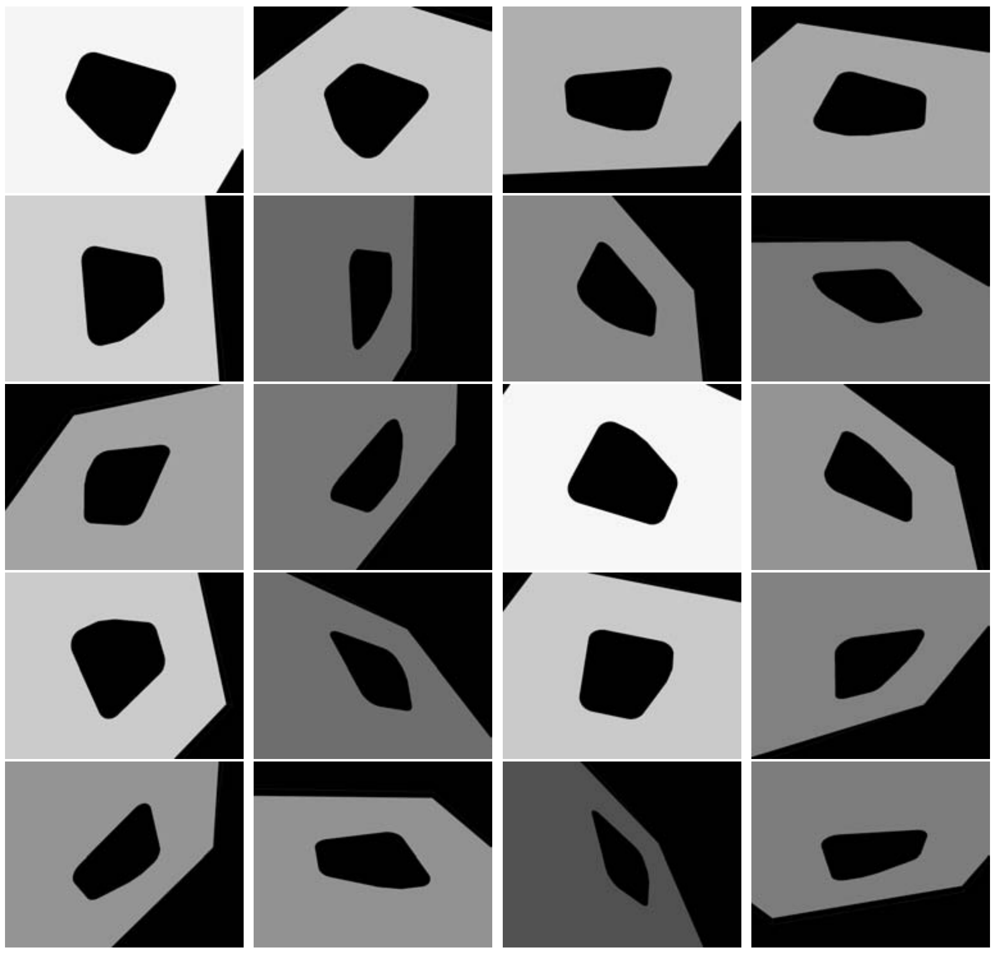


Figure 4.1: A set of images of a virtual object used to test the capacity of the algorithm to reconstruct 3D curves. The target curve is the central curve of the object.

Due to the absence of noise in the calibration and in the images, this setting should allow us to obtain a theoretical limit of our algorithm. Note that the performance is related to the complexity of the images, parallel edges close to the target curve and occlusions of parts of the curve, none of which is present in the current setting. The virtual setting will also be useful for tuning parameters related to the optimization and for testing different alternatives for the complexity control. The evaluation of the algorithm is done by measuring the distance from a set of sampled points from the estimated curve to the target model curve.

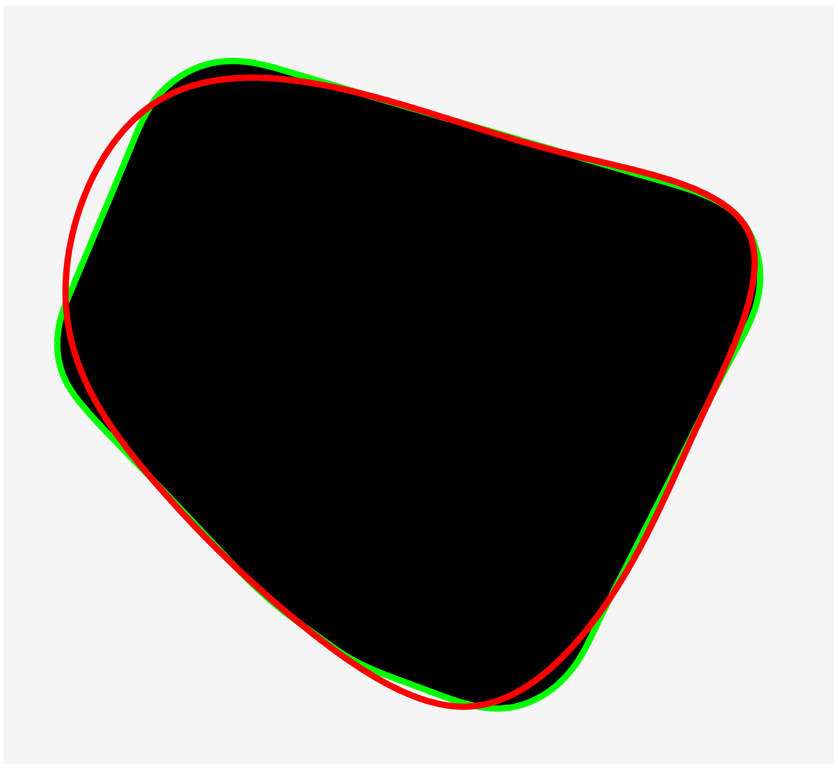


Figure 4.2: A close-up of the target curve (in green), together with the initialization used for the reconstruction algorithm (in red).

Influence of the Optimization Method

In order to compare the two optimization methods implemented, we have performed tests using the same set of images and parameters. We obtain the following results:

	Energy	Distance
Rms error	0.0323 mm	0.0796 mm
Mean error	0.0514 mm	0.113 mm
Standard deviation	0.0401 mm	0.0798 mm
Image rms error	0.175 pixels	0.309 pixels

As expected, the energy method outperforms the distance method in precision. This confirms our choice to use the energy-based optimization method in our algorithm. The distance-based method is however useful, since it converges even from a poor initialization. Using the distance function for a preliminary optimization decreases the final 3D error with approximately 10% in this setting.

We note that the error corresponds to less than a pixel in the images, which indicates that the image processing techniques used allow for a sub-pixel image precision. However, the image error being an estimation based on the edge points given by the algorithm itself, it should not be considered as an independent quality measure. Combined with the approximate scale, given by the size of a pixel at the mean distance from the object, measuring 0.22 mm, this gives at hand an image error of the same order of magnitude as the 3D error.

Influence of the Number of Views

Our set of images show the object from different viewpoints, theoretically giving enough information to reconstruct the target curve. Since there is no noise added, all images supply correct additional information concerning the 3D location of the observed curve. Nevertheless, the discrete nature of the images and the process of extraction of information from the images together with the optimization method used engender a 3D error. Therefore, increasing the number of views used in the reconstruction algorithm should increase the precision. We perform a series of tests to evaluate the influence of the number of views. We obtain the following results:

	20 images	15 images	10 images
Rms error	0.0323 mm	0.0572 mm	0.0747 mm
Mean error	0.0514 mm	0.0764 mm	0.0909 mm
Standard deviation	0.0401 mm	0.0508 mm	0.0519 mm
Image rms error	0.175 pixels	0.209 pixels	0.209 pixels

Results are given for the whole set of images, as well as for random subsets (no selection is performed) of varying sizes. For the subsets, the results given correspond to one test, but similar results have been obtained using other subsets of the same sizes.

We conclude that the effect of increasing the number of randomly chosen views is indeed a better reconstruction. However, a judicious choice of the viewpoints can reduce the impact of their number. This is even more important when occlusion occurs; The number of images must allow us to view correctly the whole curve.

Evolution of the Control Points

In order to visualize the evolution of the control points, we consider the hinge points. A hinge point is the point on the curve corresponding to a knot point. For a closed curve, there exist a one-to-one correspondence between the knot points and the control points, except for an overlap assuring the desired continuity.

The initial parametrization of the curve, based on 10 control points, is not sufficient for a description of the curve with the desired precision. Following our algorithm, new control points will therefore be inserted, until the precision is deemed good enough. In order to maximize the benefit of each inserted point, it will be placed in the interval presenting the largest image error, as described in Sec. 3.4. The development of the hinge points is shown in figure 4.3 and 4.4. For simplicity, we present only one view of the curve, as it is enough to present the insertion process. The first image shows the initial curve with its 10 hinge points. The second shows the curve after an optimization of the 3D positions of the existing control points. Since we know that the shape of the curve requires more parameters, we opt for a block insertion, adding several points at a time. For simplicity, we choose to insert one point per interval, that is, we double the number of control points. The following images show the evolution, inserting one control point at a time. We note that the hinge points tend to gather in regions of higher curvature, such as corners and bends. As the image error, the mean error as well as the rms error, is less than a pixel after just one iteration, the difference between two iterations is hardly visible.

Complexity Control

In our evaluation setting, the description of the curve that we observe in the images is known. We can therefore, in each step of the algorithm, compare the current estimate to the ground truth. The evolution of this 3D error, together with the 2D image error, are presented in figure 4.5, whereas AIC and BIC are displayed in figure 4.6. We note that the 3D error and the 2D image error evolve in a similar way, as do AIC and BIC. Since the AIC and the BIC curves do not present a minimum, but decrease toward an asymptotical lower limit, they are hard to use in the decision process for the complexity control. Analogous results are found using different subsets of the images.

As the two dedicated criteria do not seem to be able to indicate the optimal complexity, we instead opt for a simpler way to decide when to stop the iterative procedure. We fix in advance an upper limit for the number of control points, based on the shape of the curve as given by the CAD model, that is, the size and the complexity. We then iterate until the image error stabilizes. A stable image error is assumed attained when the difference between iterations is below some threshold for a given number of iterations (we take 0.1 times the maximum number of iterations allowed).

We recall that once the optimal complexity is fixed, a final minimization

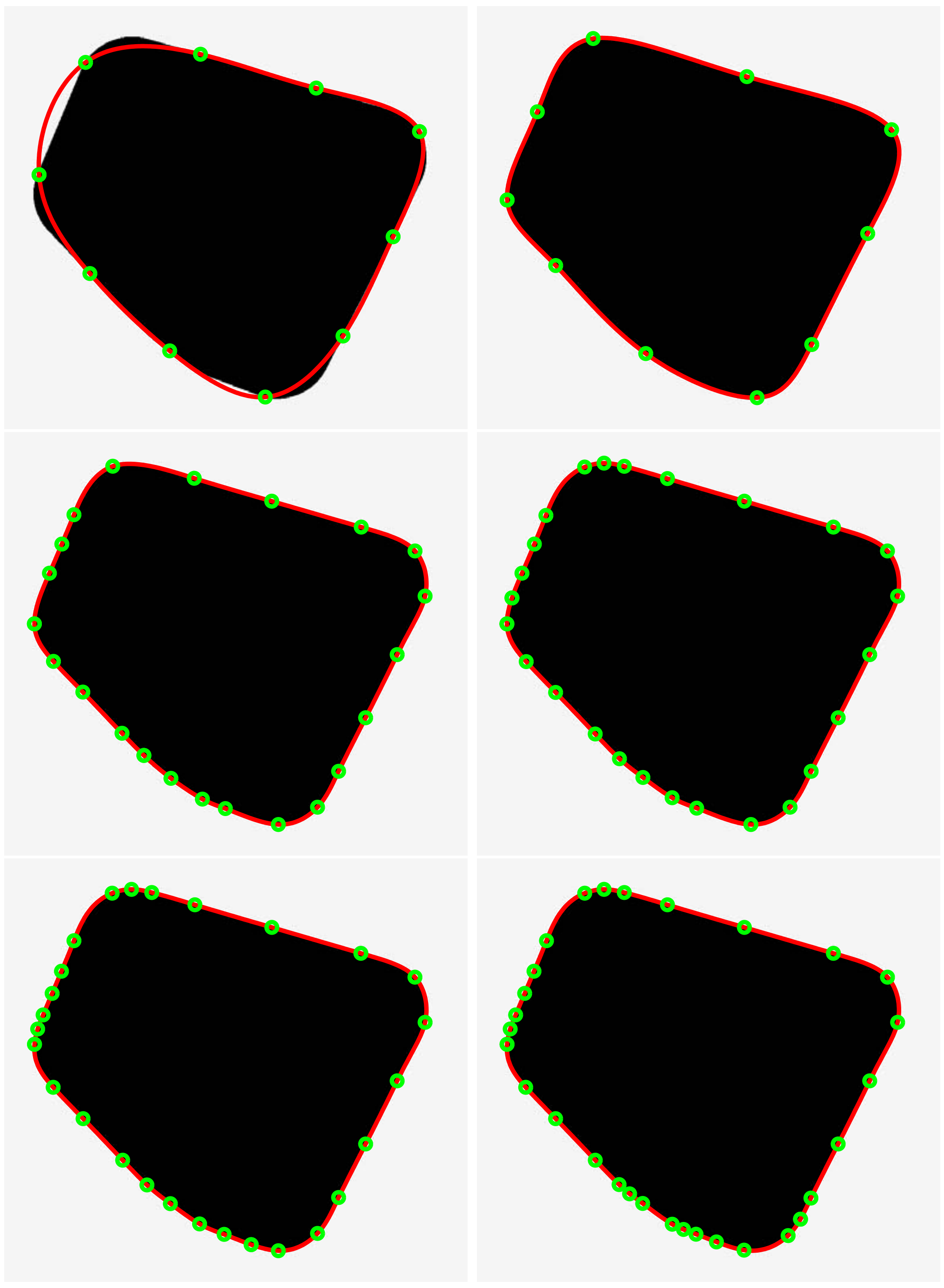


Figure 4.3: Evolution of the curve throughout the control point insertion process. The reprojection of the curve into one of the images is shown together with its hinge points, that is, the locations on the curve corresponding to a knot point. We show approximately one iteration out of two.

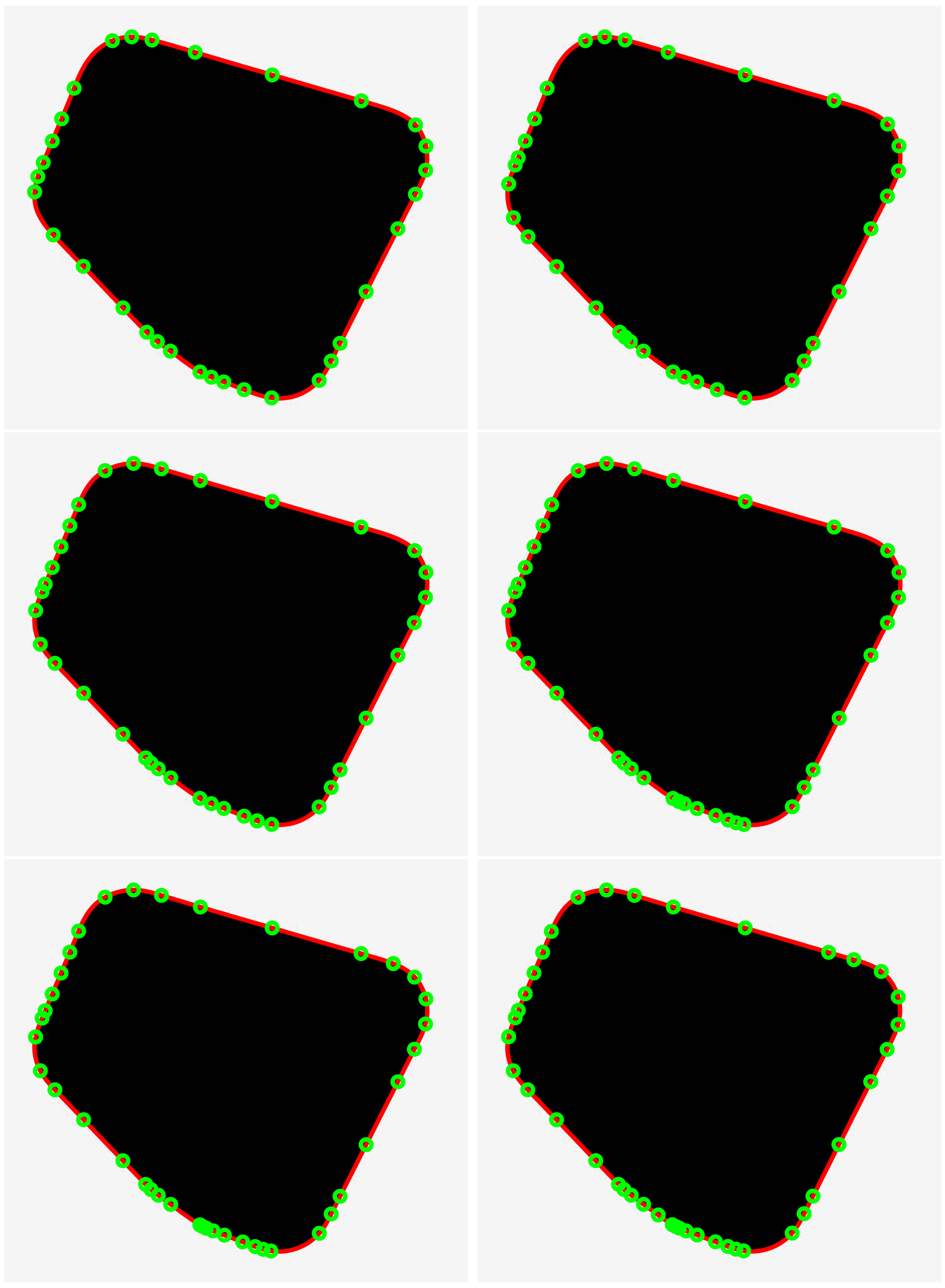
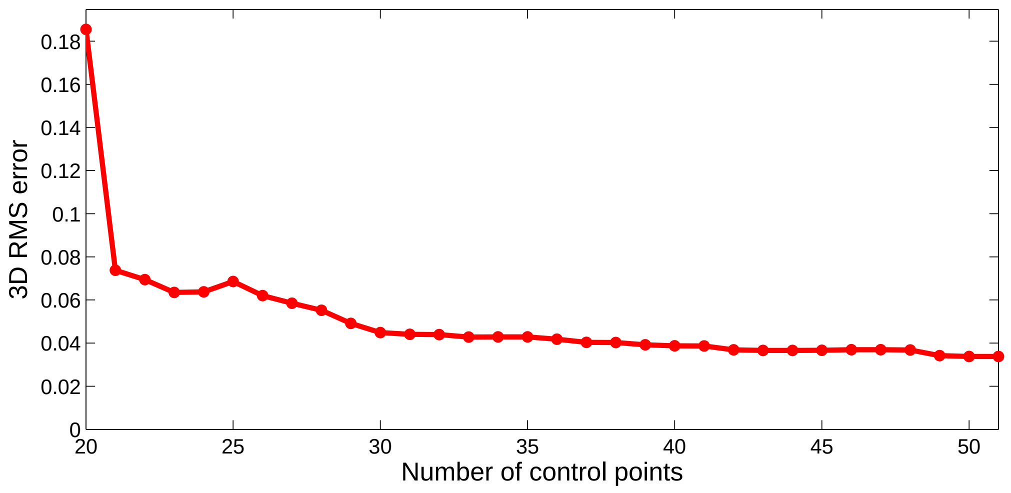
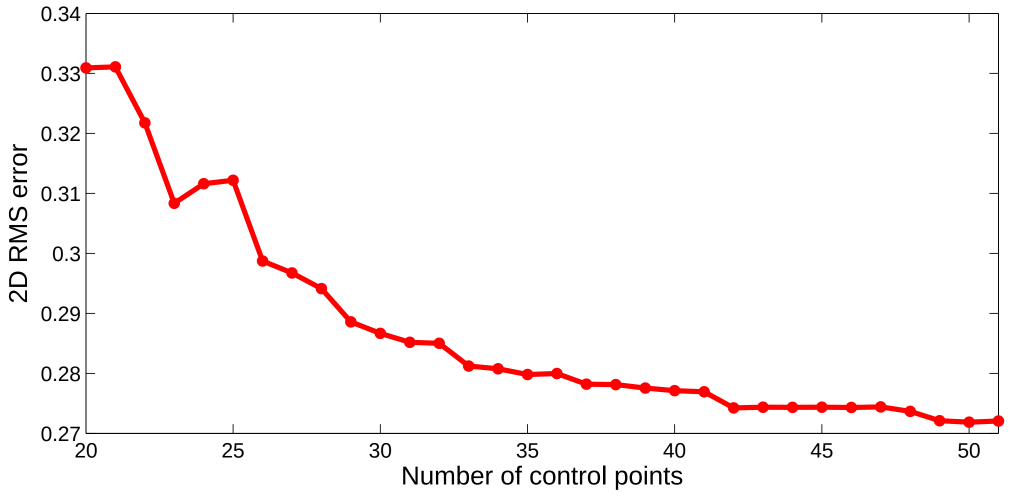


Figure 4.4: Evolution of the curve (continued).



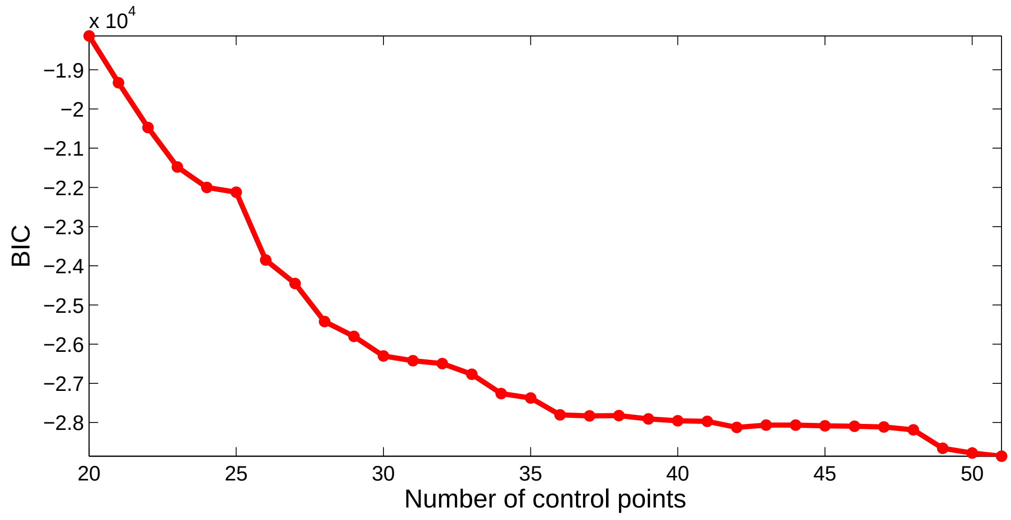
(a)



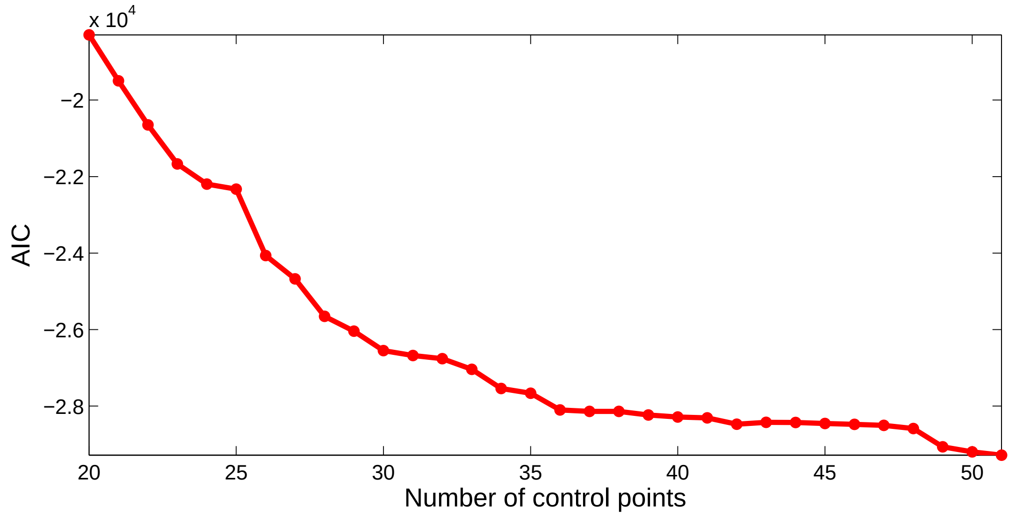
(b)

Figure 4.5: Evolution of the 3D rms error and the image rms error throughout the control point insertion process.

over all control points and the external camera matrices for each one of the views is performed. In the case of virtual images with perfectly known camera parameters, the latter is evidently redundant.



(a)



(b)

Figure 4.6: Evolution of BIC and AIC throughout the control point insertion process..

4.3 Real Images

4.3.1 Standard Reference Object

Setting

A second series of tests has been performed on a set of real images of a standard reference object, see figure 4.7, known to a very high precision (incertainty of 10 μm). The image size is 1300×1030 pixels and the object is seen from a large distance, compared to its size. We again use a single, closed target curve, shown in figure 4.8. The initialization is a simplified version of the model curve, but rather close to the original. The starting curve has 5 control points, which theoretically should be enough to describe a circle. The diameter of the circle is 30 mm, which corresponds roughly to 30 pixels at the mean view distance. For the testing, we use subsets of 8 images.

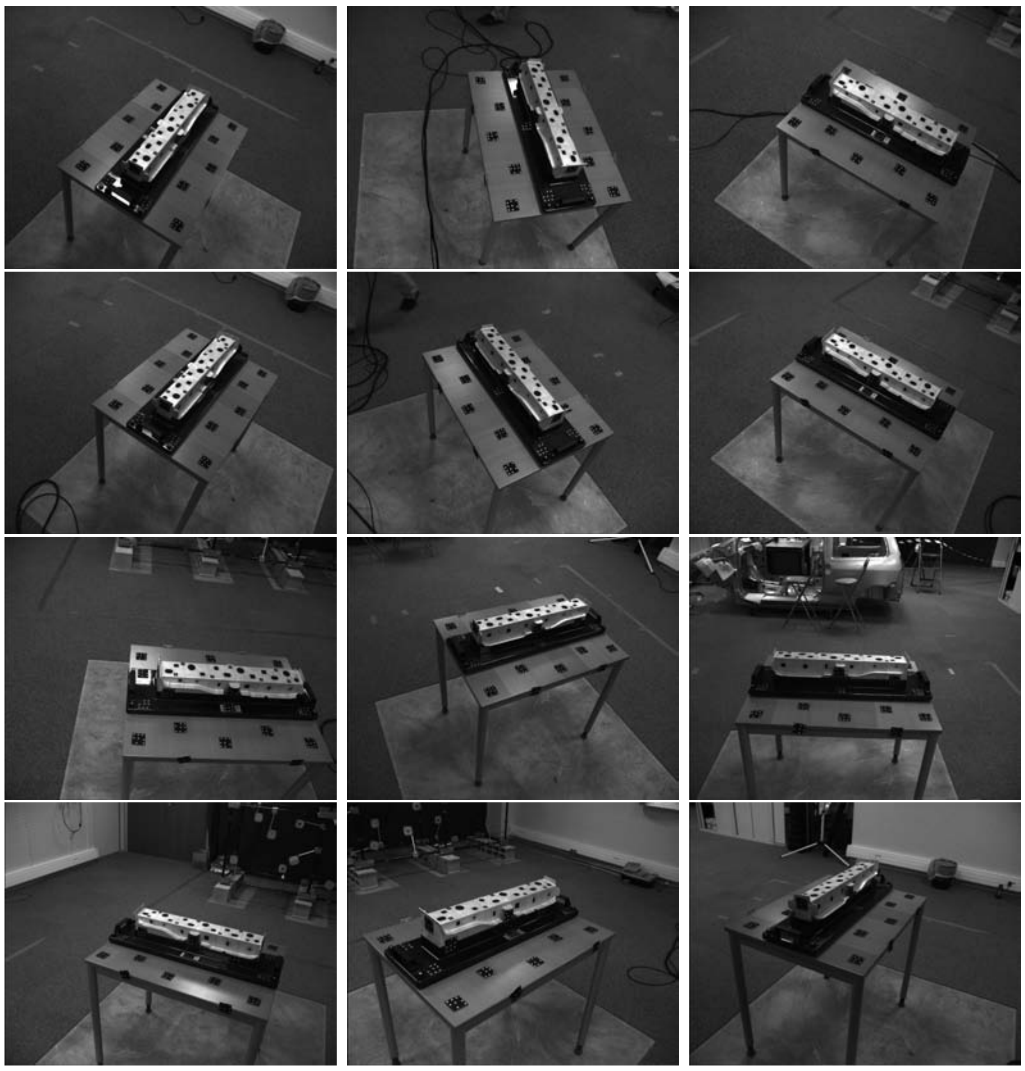


Figure 4.7: A set of images of a standard test object used for the 3D reconstruction of the object curves.

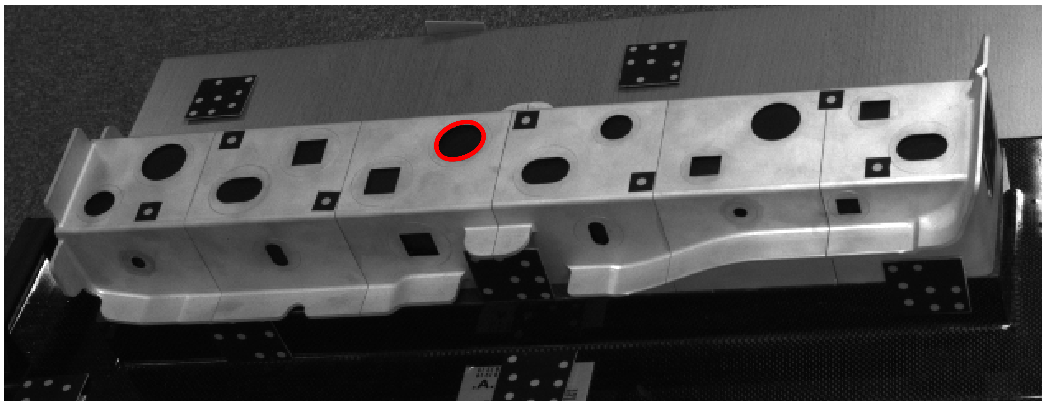


Figure 4.8: A standard reference object and the target curve used for our computations. The curve is a circle and can thus be described exactly by a non-rational curve.

Results

The reprojections of the reconstructed 3D curve are shown in figure 4.9. We obtain the following errors:

	Subset 1	Subset 2	Subset 3
Rms error	0.127 mm	0.137 mm	0.121 mm
Mean error	0.143 mm	0.146 mm	0.133 mm
Standard deviation	0.0642 mm	0.0518 mm	0.0570 mm
Image rms error	0.0745 pixels	0.0827 pixels	0.0833 pixels

Due to the viewing distance, the image quality and the noise in the camera parameters, the errors are somewhat higher than in the case of virtual images. The errors given correspond to the results after a final optimization over all control points and the external camera parameters for the entire set of images. In order to show the effect of the final optimization, we present, for the same subsets, the rms errors before and after this last step in the algorithm:

	Subset 1	Subset 2	Subset 3
Rms error (before)	0.166 mm	0.149 mm	0.148 mm
Rms error (after)	0.127 mm	0.137 mm	0.121 mm

4.3.2 Aeronautic Part

Setting

We consider a set of images of an aeronautic part, see figure 4.10. The image size is 1392 × 1040 pixels and the object is shown in close-up. A single, closed

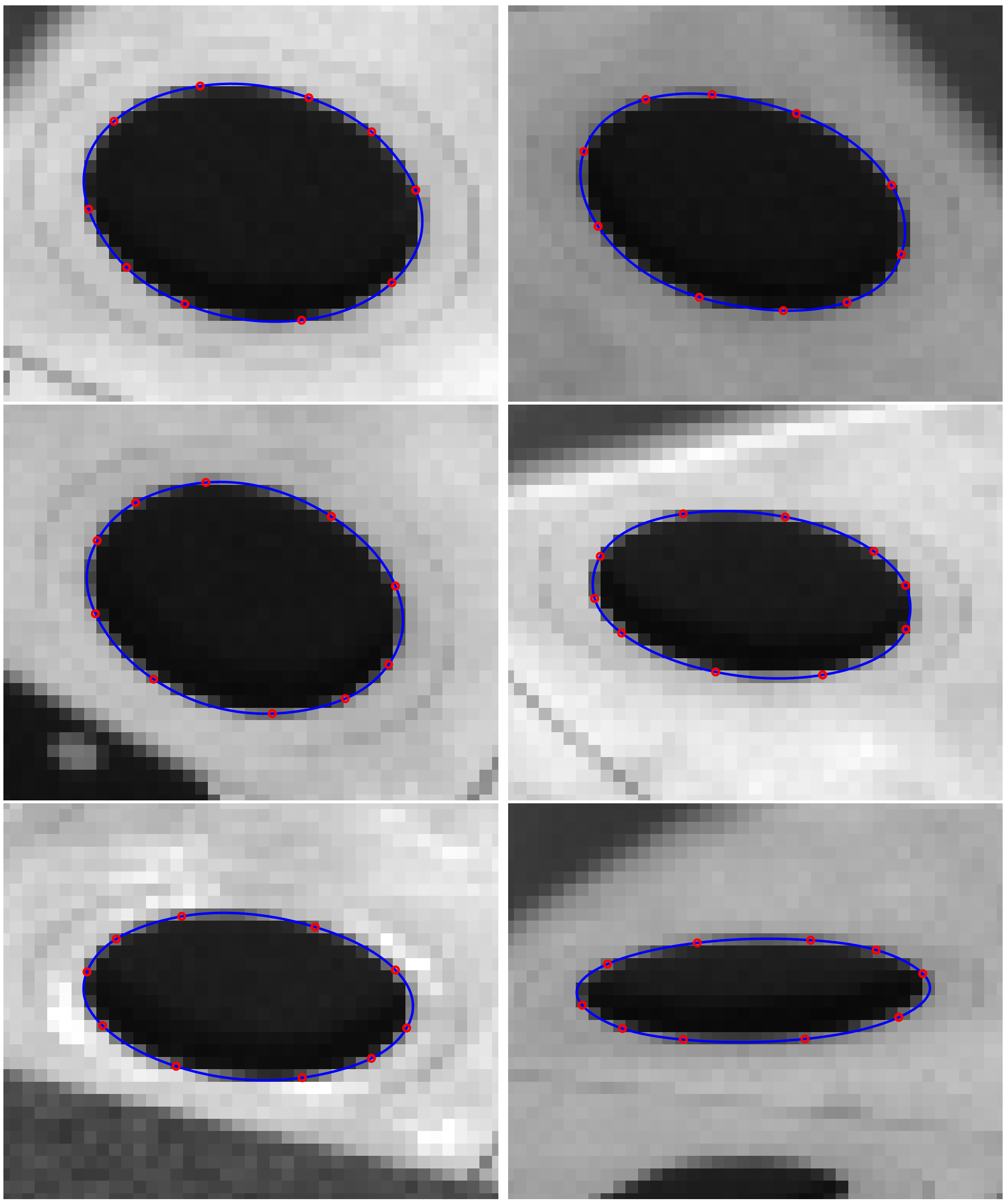


Figure 4.9: Evolution of the curve throughout the control point insertion process. Reprojections of the reconstructed 3D curve into the images, together with its hinge points.

target curve is used at a time. One closed and one open curve are chosen for the testing. They are shown, together with modified ‘model curves’ used as initialization in figure 4.11 and figure 4.12. The starting curves have 10 and 8 control points respectively, which is not enough to correctly describe the curves correctly. The initial sampling used for the computations is of 200 and 160 points respectively. At the mean distance from the object curve, one pixel corresponds roughly to 0.28 mm. The largest dimension of the closed curve corresponds approximately to 400 pixels. The computations are based on the complete series of 36 images.

We need to face the problem of noisy image data, multiple parallel structures and imprecision in the localization and the calibration of the views.

Results

The evolution of the hinge points for the closed curve is shown in figure 4.13 and for the open curve in figure 4.14 and figure 4.15. We obtain the following reconstruction errors:

	Closed curve	Open curve
Rms error	0.115 mm	0.105 mm
Mean error	0.099 mm	0.125 mm
Standard deviation	0.076 mm	0.0669 mm
Image rms error	0.366 pixels	0.304 pixels

The errors are similar for the two target curves that are of approximately the same length. The use of open or closed curves is indifferent for our algorithm. We note that the higher error of the edge detection, partly due to a complex environment in the images, induce a higher reconstruction error. As before, recalling the size of a pixel at the mean distance from the object, 0.28 mm, gives us an approximate image error close to the 3D error.

In order to confirm the results obtained using the simplified model, we have tested the two different optimization methods implemented. Also in the case of real images, the energy-based method yields a slightly better result. We obtain the following errors:

	Energy	Distance
Rms error	0.115 mm	0.12 mm
Mean error	0.099 mm	0.149 mm
Standard deviation	0.076 mm	0.0889 mm
Image rms error	0.366 pixels	0.361 pixels

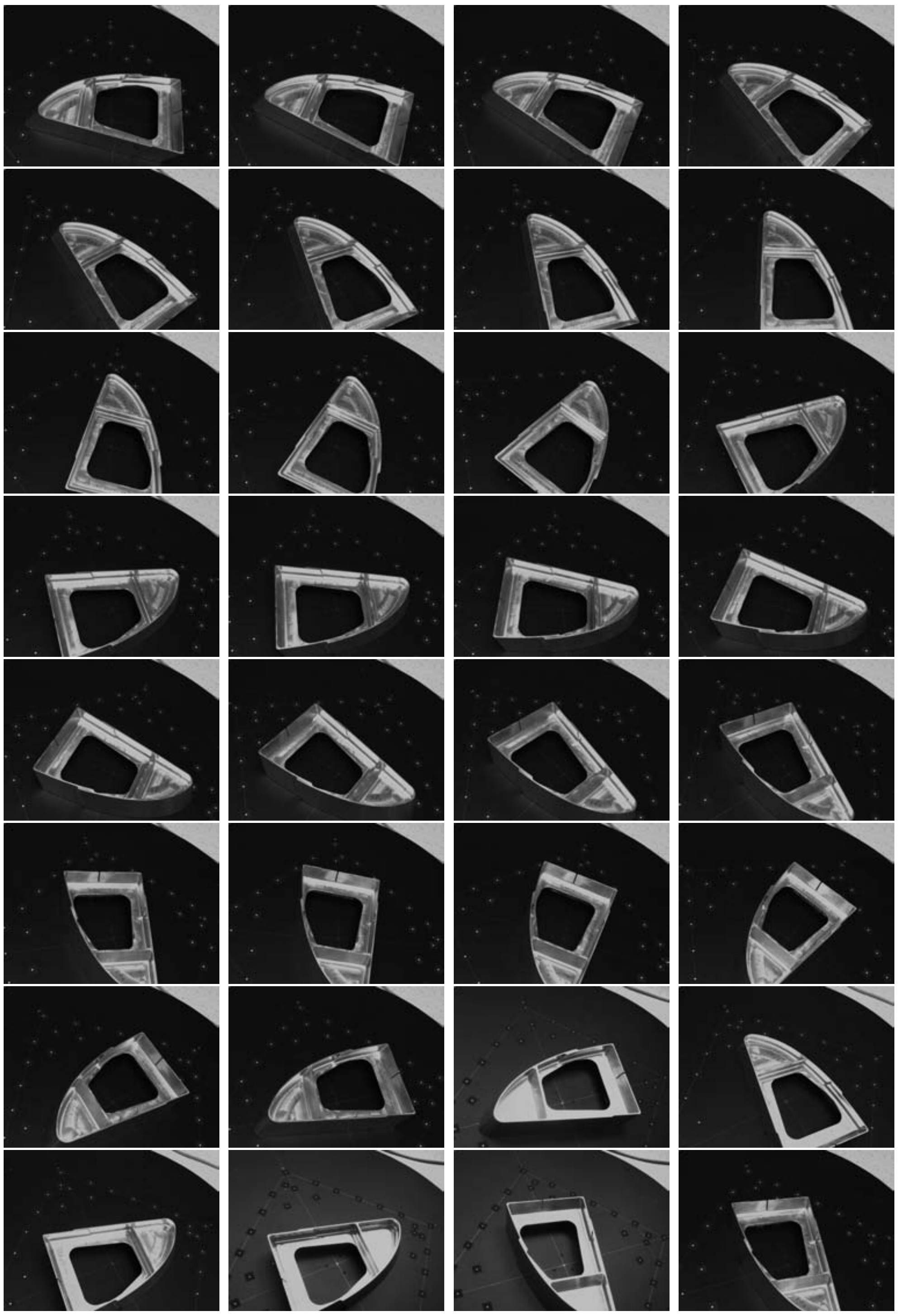


Figure 4.10: A set of images of an aeronautic part used for the reconstruction of the object curves.

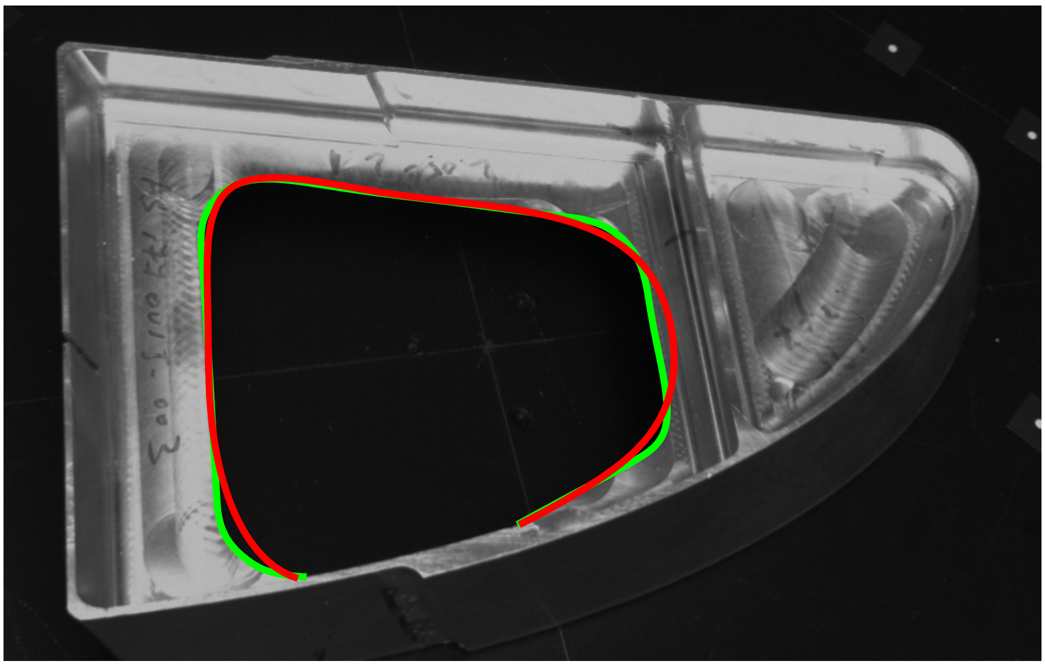


Figure 4.11: A closed target curve used in our evaluation. The initialization is shown in red, whereas the model curve is shown in green.

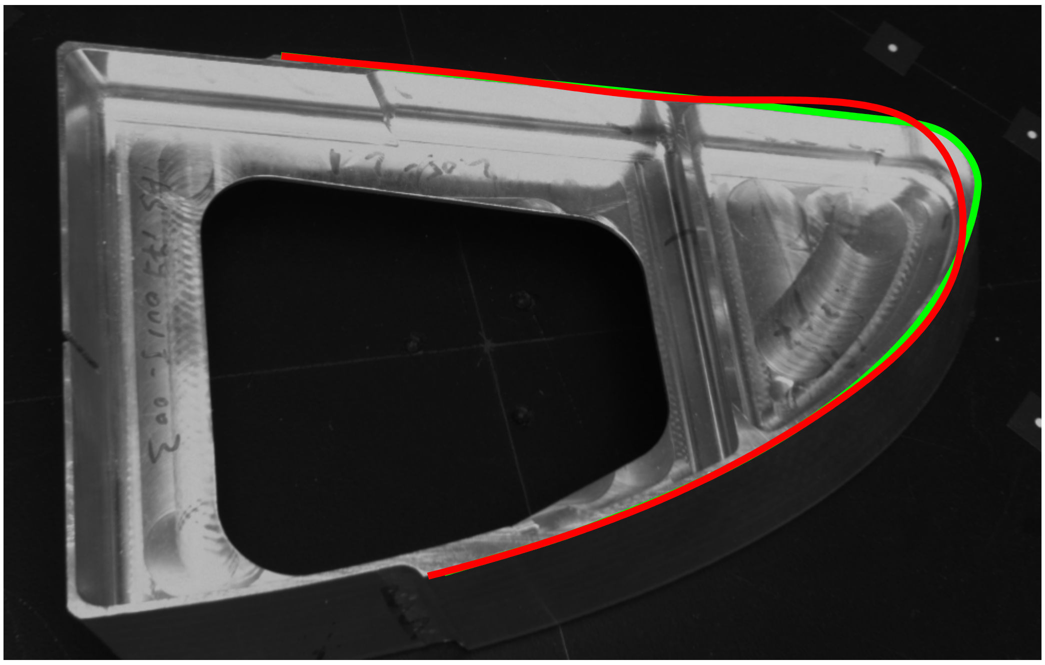


Figure 4.12: An open target curve used in our evaluation. The initialization is shown in red, whereas the model curve is shown in green.

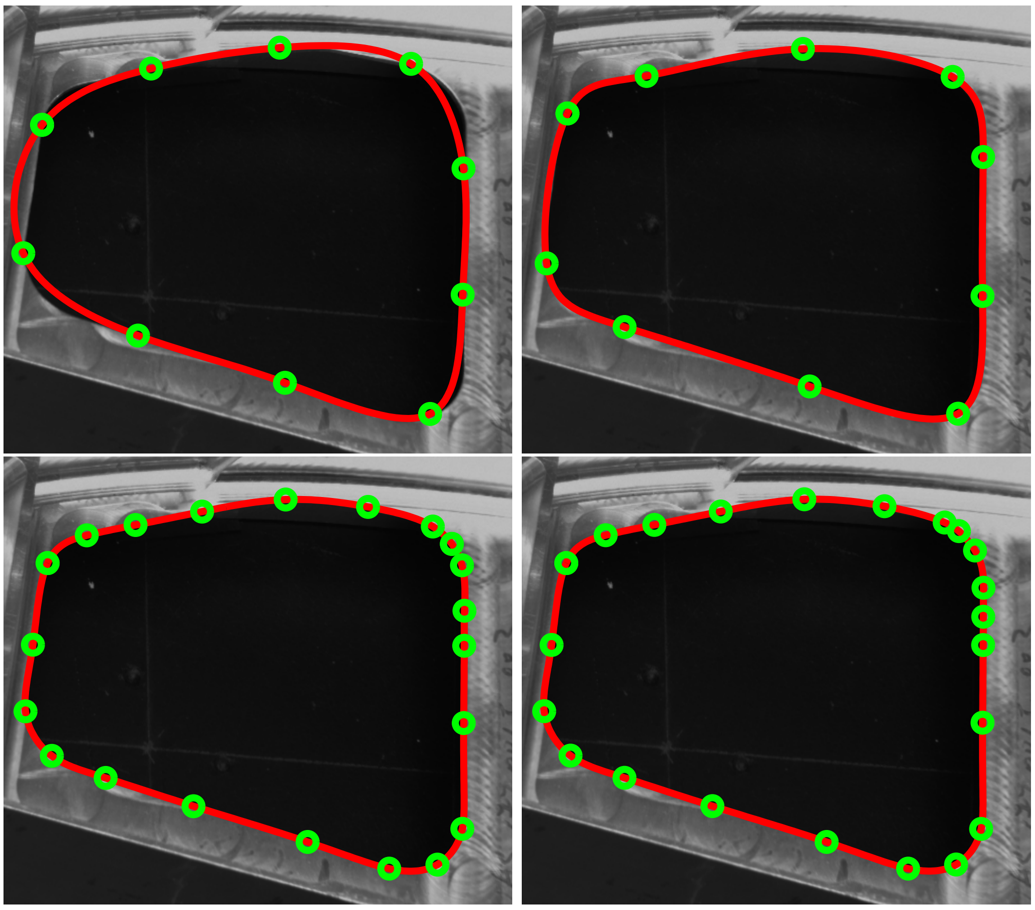


Figure 4.13: Evolution of the curve throughout the control point insertion process. The reprojection of the curve into one of the images is shown together with its hinge points, that is, the locations on the curve corresponding to a knot point.

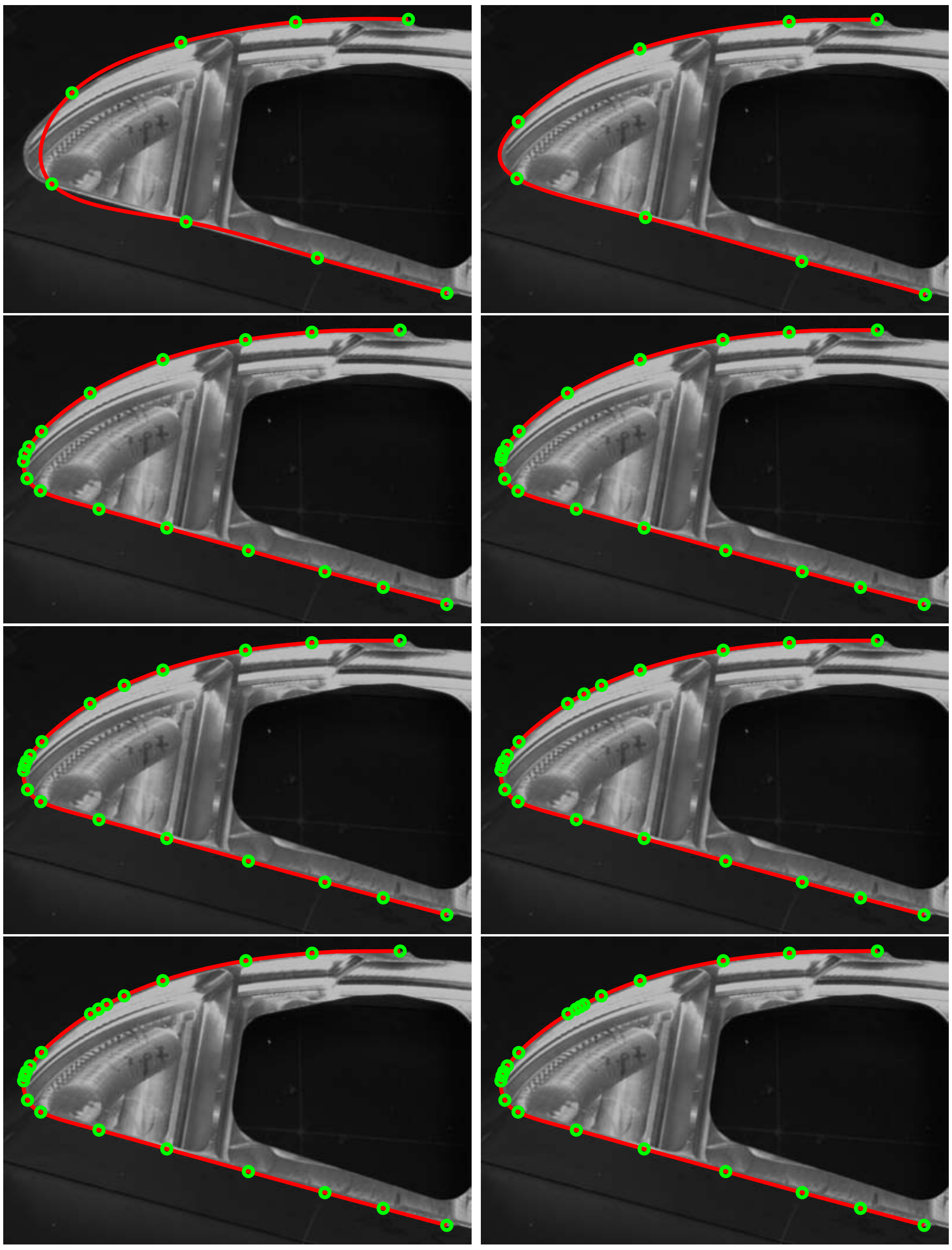


Figure 4.14: Evolution of the curve throughout the control point insertion process. The reprojection of the curve into one of the images is shown together with its hinge points, that is, the locations on the curve corresponding to a knot point.

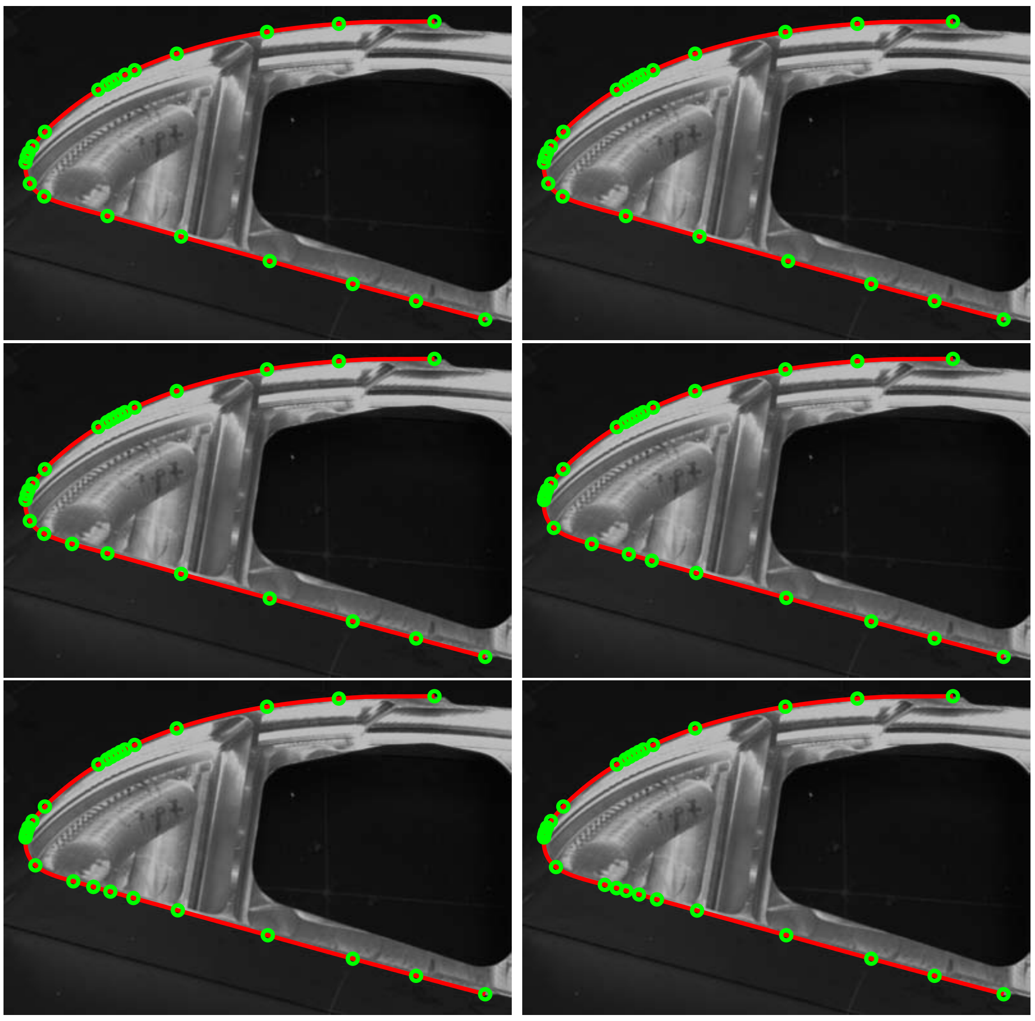


Figure 4.15: Evolution of the curve and its hinge points throughout the iterations (continued).

4.4 Nonconformities

Setting

We have created a set of 27 virtual images created using the CAD model of the aeronautic part used in Sec. 4.3.2. In order to simulate a deformation of the target object, we have modified the model, adding a bump to an otherwise straight edge, prior to creating the virtual images. The initialization is the same as in the case of the simplified virtual images and the closed curve of the aeronautic part. The image size is 1494 × 1078 pixels.

Results

Based on different subsets of 5 images each and starting at the undeformed model curve, our algorithm manages to reconstruct the curve and its anomaly. The results are displayed in figure 4.17 and figure 4.18. We obtain the following errors:

	Set 1	Set 2	Set 3	Set 4	Set 5
Rms error (mm)	0.0675	0.0809	0.0591	0.0694	0.0639
Mean error (mm)	0.0798	0.0944	0.0779	0.0826	0.0757
Standard deviation (mm)	0.0427	0.0487	0.0508	0.045	0.0406
Image rms error (pixels)	0.186	0.195	0.24	0.192	0.202

We note that the errors are only slightly higher than in our first test setting using a simplified virtual object. Although the reconstruction is good, the error is concentrated around the anomaly, which is somewhat smoothed out. Note that the errors are measured with respect to the known, deformed curve. Since the set of views of the object is chosen based on the actual curve, any view will be useful for the reconstruction. Nevertheless, the images where the anomaly is less visible will not add much information on this critical part of the curve and the effect of using all images will thus be a loss in precision. Subsets of this size cannot be random, but must be picked either manually or using an automatic strategy [37].

4.5 Algorithm

The different tests performed during the evaluation of our method have guided the development. The choices made have been validated by the results, using virtual as well as real images. An outline of the hybrid algorithm, taking advantage of the distance minimization as well as the energy minimization, is given in Tab. 4.1. The principal choices retained concern the hybrid

```
ALGORITHM
• Visibility Check Identification of the visible parts
• Line-search for contour points $\mathbf{q}_{ij}$ matching $\mathbf{p}_{ij} = \sum_{l=0}^{n-1} T_i(\mathbf{P}_l)\mathbf{R}_{l,k}^{(i)}(t_j)$
• Optimization on all control points: minimization of the residual 2D error
• Optimization on all control points: minimization of an energy functional
• Computation of the $rms_{2D}$
• do (control point insertion)
    ◦ Line-search for contour points $\mathbf{q}_{ij}$
    ◦ Computation of the median error $m_K$ for each interval $I_K$.
    ◦ Knot point insertion at the midpoint of interval $I = \arg\min_K m_K$.
    ◦ Visibility Check Identification of the visible parts
    ◦ Optimization on the control points linked to the identified interval:
      minimization of an energy functional
    ◦ Computation of the $rms_{2D}$
• while ($rms_{2D}$ not stabilized)
• Optimization on all control points and the pose parameters: minimization
  of an energy functional
```

Table 4.1: Hybrid reconstruction algorithm, using residual 2D error minimization as well as energy functional minimization.

algorithm, the refinement of the control point positions and the external camera parameters, the adaptive sampling, the control point insertion and the stopping criterion.

4.6 Conclusions

Through several series of tests, we have evaluated the performance of our curve reconstruction algorithm. Tests on a simplified virtual object have allowed us to validate strategic choices in our procedure, such as the optimization method and the stopping criterion for he complexity control. They have also allowed to measure the influence of the number of views used. Tests on industrial objects, using different target curves, have confirmed the ability of the algorithm to reconstruct curves observed in real images. Although a somewhat higher precision is required for industrial applications, the results are encouraging. Finally, tests on virtual objects presenting nonconformities have shown that even a minor defect is detected by the algorithm. As we have included no automatic interpretation of the reconstructed curve, 'detected' should be understood in the sense 'correctly reconstructed'.

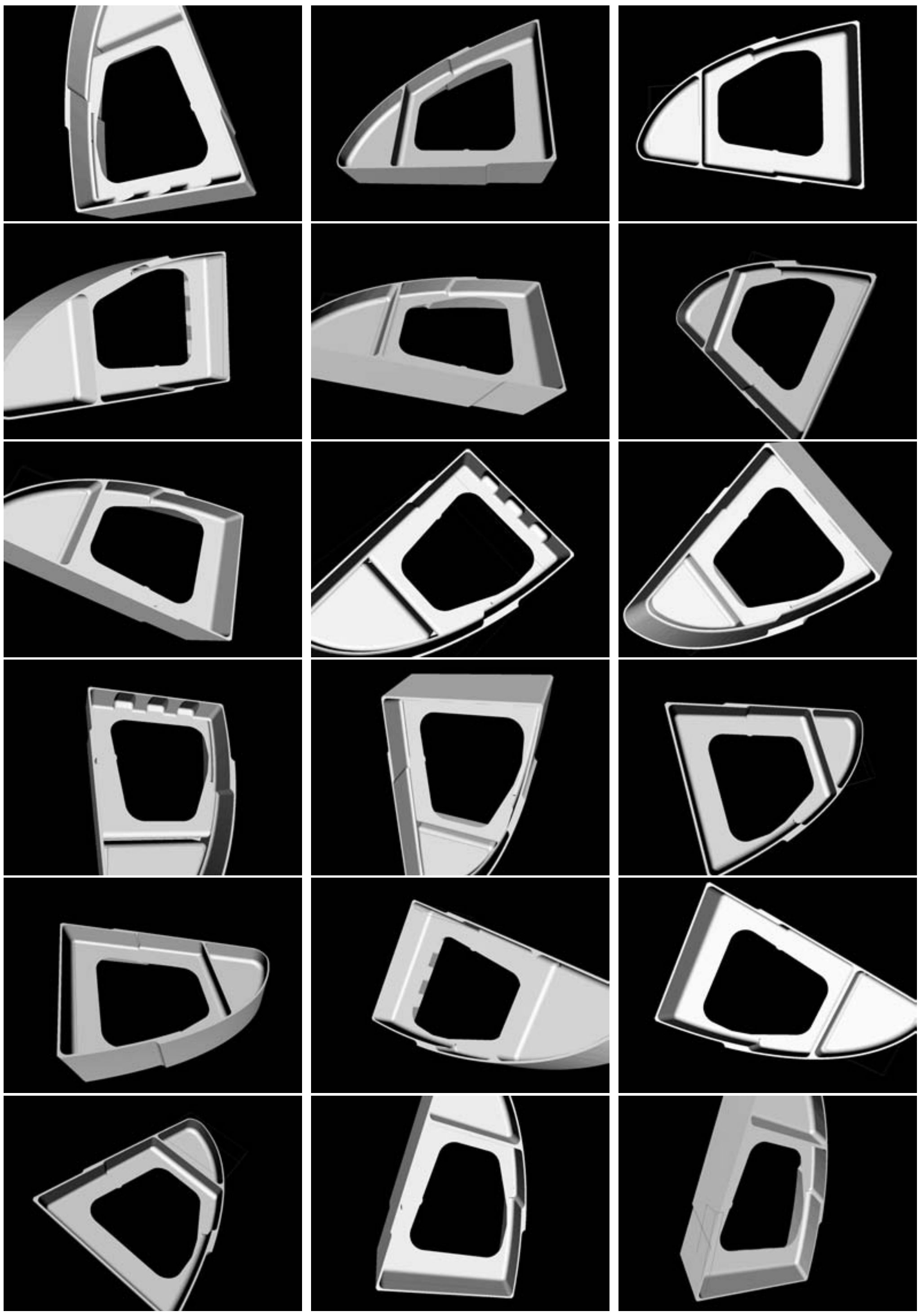


Figure 4.16: A subset of the series of virtual images of an object presenting a nonconformity.

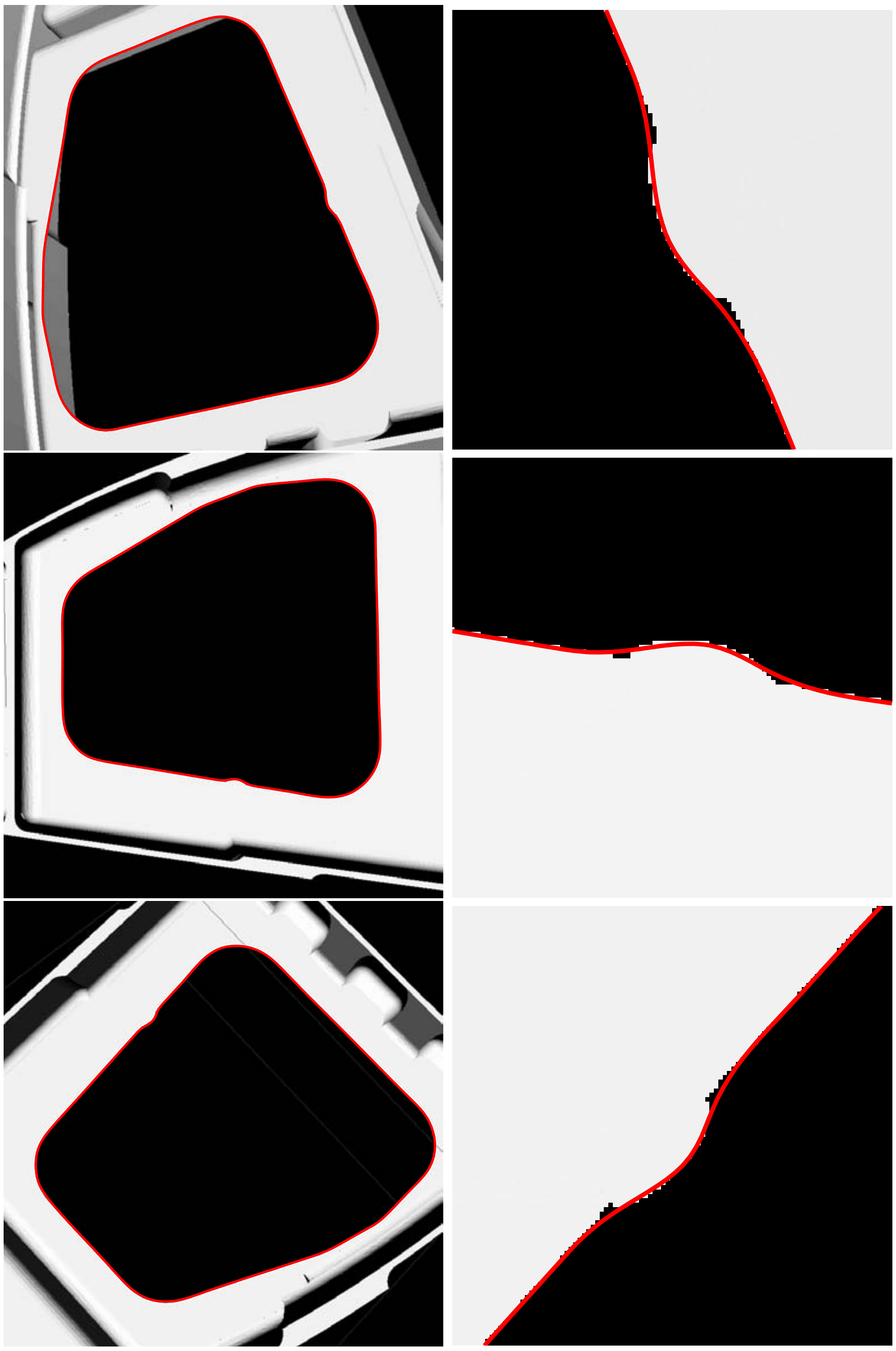


Figure 4.17: Reconstruction of a nonconformity.

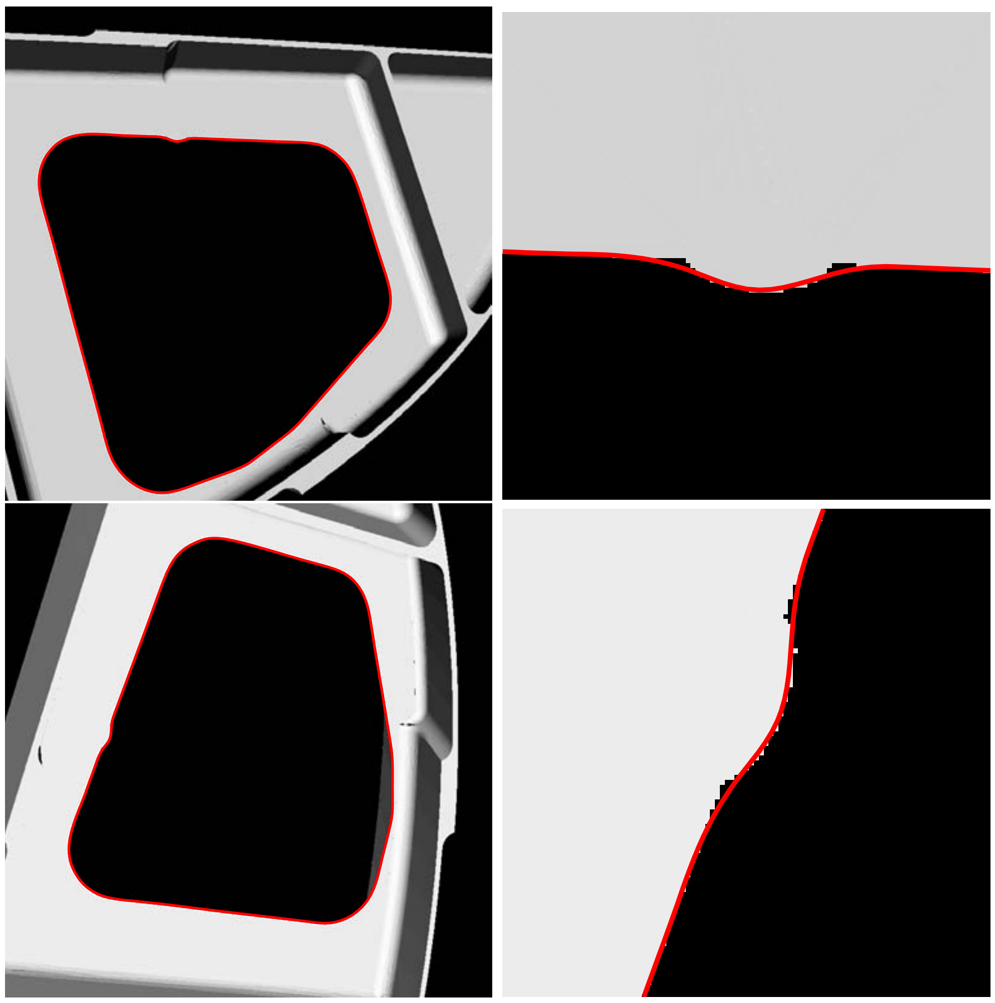


Figure 4.18: Reconstruction of a nonconformity (continued).

Conclusions and Perspectives

In this thesis, we have developed methods for the 3D reconstruction of objects in the context of quality control in the manufacturing industry. The constraints related to this field are strong: the objects are in general metallic, which causes a specular reflections in the images; the complexity of the objects leads to images with a high information density; the structure of the objects is strongly 3D, yielding important self-occlusions. Moreover, the applications that we target require a high precision in order to challenge existing methods.

We have presented two building blocks in the process of 3D reconstruction of industrial objects. Our first contribution is in the field of affine point reconstruction. We have developed a method that allows the alignment of two or more affine reconstructions in the presence of missing data. The reconstruction thus obtained is an approximation that is intended to be used as an initialization for a more precise, nonlinear method, such as bundle adjustment [65]. The procedure that is given covers the reconstruction of a point cloud, thereby being appropriate for the surface elements. In order to describe the discontinuities of the objects, such as edges and holes, we have considered the reconstruction of curves. In a second part of our work, we deal with the reconstruction of parametric curves taken from a CAD model of the object. We have proposed an algorithm that extracts a curve, reprojects it in the images and reconstructs the curve observed via the minimization of a cost function based on 2D image data. The parameterization is adapted, so as to fit the observed curve with its possible anomalies.

The algorithms developed have been evaluated experimentally, using synthetic as well as real image data. The results of the evaluation are encouraging and show that techniques from computer vision are well suited for applications in the field of quality control.

Further work need to be done in order to develop a complete system that is able to handle the entire object. The simultaneous reconstruction of several curves and surface elements would yield a more robust behavior and a consistent description of the objects and its anomalies. Furthermore, an automatic, application-specific interpretation process that is able to label detected irregularities is needed to complete the pure reconstruction of the observed object. Moreover, the knowledge of possible anomalies, that are more or less likely to appear during the assembly, could also be incorporated into the reconstruction procedure and serve to guide it.

Conclusions et perspectives

Dans le cadre de cette thèse, nous avons développé des méthodes pour la reconstruction 3D d'objets dans le contexte du contrôle dimensionnel de production. Les contraintes liées à ce domaine sont fortes : Les objets sont souvent métalliques, ce qui cause des problèmes de spécularités dans les images ; la complexité des objets donne des images denses en information, donc difficiles à traiter ; la structure des objets est fortement 3D, ce qui donne lieu à de nombreuses auto-occultations. En outre, les systèmes de contrôle sans contact à partir de vision doivent atteindre une très grande précision pour respecter les contraintes requises par l'industrie pour le contrôle dimensionnel de production.

Nous avons présenté deux briques pour le procédé de la reconstruction 3D d'objets industriels. Notre première contribution se situe dans le domaine de la reconstruction affine de points à partir d'images. Nous avons développé une méthode qui permet d'aligner deux ou plusieurs reconstructions affines en présence de données manquantes. La reconstruction ainsi obtenue reste une approximation de la reconstruction 3D, destinée à servir à initialiser une méthode non-linéaire, plus précise, telle que l'ajustement de faisceaux [65]. La procédure donnée reconstruit un nuage de points et est en conséquence adaptée aux surfaces.

Afin de décrire les discontinuités de l'objet, telles que les contours et les trous, nous avons considéré la reconstruction 3D de courbes. Dans une deuxième partie de nos travaux, nous traitons la reconstruction de courbes paramétriques à partir du modèle CAO de l'objet. Nous avons proposé un algorithme qui extrait une courbe et la reconstruit par la minimisation d'une fonction de coût basée sur les données image. La paramétrisation est adaptée afin d'être en adéquation avec la courbe observée et ses éventuelles anomalies.

Les algorithmes développés ont été évalués expérimentalement, par le biais de données de synthèse ainsi que d'images réelles. Les résultats de l'évaluation montrent que les techniques tirées de la vision par ordinateur sont bien adaptées pour les applications dans le domaine du contrôle de conformité industriel.

Des futurs travaux permettront de développer un système capable de traiter l'objet entier. La reconstruction simultanée des courbes et des éléments de surface permettrait d'accroitre la robustesse de la méthode et de donner une description cohérente de l'objet avec ses anomalies. En outre, l'intégration d'un procédé automatique d'interprétation, spécifique à l'application, capable de labelliser les anomalies détectées en complément de la reconstruction pure est indispensable pour un système industriel. Enfin, la connaissance des anomalies prévisibles, plus ou moins probables de survenir au cours de l'usinage, pourrait aussi être intégrée dans le procédé.

Bibliographie

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automated Control*, 19(6) :716–723, 1974.

[2] K. B. Atkinson, editor. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, 2001.

[3] N. Ayache. *Artificial Vision for Mobile Robots – Stereo-Vision and Multisensor Perception*. MIT-Press, 1991.

[4] A. Bartoli, H. Martinsson, F. Gaspard, and J-M. Lavest. On aligning sets of points reconstructed from uncalibrated affine cameras. In *14th Scandinavian Conference on Image Analysis*, pages 531–540, Joensuu, Finland, 2005.

[5] B. Bascle and R. Deriche. Stereo matching, reconstruction and refinement of 3D curves usingdeformable contours. In *4th International Conference on Computer Vision*, pages 421–430, Berlin, Germany, 1993.

[6] P.A. Beardsley, A. Zisserman, and D.W. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3) :235–259, 1997.

[7] S. Brandt. Closed-form solutions for affine reconstruction under missing data. In *Statistical Methods for Video Processing Workshop*, pages 109–114, 2002.

[8] P. Brigger, J. Hoeg, and M. Unser. B-spline snakes : A flexible tool for parametric contour detection. *IEEE Transactions on Image Processing*, 9(9) :1484–1496, July 2000.

[9] C. Canero, P. Radeva, R. Toledo, J.J. Villanueva, and J. Mauri. 3D curve reconstruction by biplane snakes. In *15th International Conference on Pattern Recognition*, volume 4, pages 563–566, 2000.

[10] T.-J. Cham and R. Cipolla. Stereo coupled active contours. In *Conference on Computer Vision and Pattern Recognition*, pages 1094–1099. IEEE Computer Society, 1997.

[11] T.-J. Cham and R. Cipolla. Automated B-spline curve representation incorporating MDL and error-minimizing control point insertion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1) :49–53, January 1999.

[12] G. Csurka, D. Demirdjian, and R. Horaud. Finding the collineation between two projective reconstructions. *Computer Vision and Image Understanding*, 75(3) :260–268, 1999.

[13] F. Dellaert, S. M. Seitz, C. Thorpe, and S. Thrun. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 50(1-2) :45–71, 2003.

[14] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Ser. B*, 39(1) :1–38, July 1977.

[15] P. Dierckx. *Curve and Surface Fitting with Splines*. Oxford University Press, Inc., New York, NY, USA, 1993.

[16] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7 :932–946, 2002.

[17] O. Faugeras. *Three-Dimensional Computer Vision : A Geometric Viewpoint*. MIT Press, 1999.

[18] O. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images*. MIT Press, 2001.

[19] M. Figueiredo, J. Leitão, and A.K. Jain. Unsupervised contour representation and estimation using B-splines and a minimum description length criterion. *IEEE Transactions on Image Processing*, 9(6) :1075–1087, June 2000.

[20] M.A. Fischler and R.C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6) :381 – 395, 1981.

[21] A.W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *5th European Conference on Computer Vision*, pages 311–326, 1998.

[22] J. Fryer, H. Mitchell, and J. Chandler, editors. *Applications pf 3D Measurements from Images*. Whittles Publishing, 2007.

[23] H. Fuchs, Z. Kedem, and B. Naylor. On visible surface generation by A priori tree structures. *Conference Proceedings of SIGRAPH '80*, 14(3) :124–133, July 1980.

[24] R. Ghaffari. Snake contours in three-dimensions from colour stereo image pairs. Master's thesis, School of Computing Science, Simon Fraser University, 2005.

[25] G.H. Golub and C.F. van Loan. *Matrix Computation.* The Johns Hopkins University Press, Baltimore, 1989.

[26] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the EM algorithm. In *Conference on Computer Vision and Pattern Recognition*, pages 707–714, 2004.

[27] M. H. Hansen and B. Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454) :746–774, 2001.

[28] C. Harris. Determination of ego-motion from matched points. In *3rd Alvey Vision Conference*, 1987.

[29] C. Harris and J. M. Pike. 3d positional integration from image sequences. In *3rd Alvey Vision Conference*, pages 233–236, 1987.

[30] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Computer Vision and Pattern Recognition*, pages 761–764, 1992.

[31] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision – 2nd Edition.* Cambridge University Press, 2003.

[32] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7) :1127–1135, 1988.

[33] D. Jacobs. Linear fitting with missing data : Applications to structure-from-motion and to characterizing intensity images. In *Conference on Computer Vision and Pattern Recognition*, pages 206–212, 1997.

[34] F. Kahl and J. August. Multiview reconstruction of space curves. In *9th International Conference on Computer Vision*, volume 2, pages 1017–1024, 2003.

[35] K. Kanatani. Geometric information criterion for model selection. *International Journal of Computer Vision*, 26(3) :171–189, 1998.

[36] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : Active contour models. *International Journal of Computer Vision*, 4(1) :321–331, 1987.

[37] K. Krauss. *Photogrammetry.* Dümmler Verlag, 1997.

[38] K. Levenberg. A method for the solution of certain problems in least squares. *Quarterly Applied Mathematics*, 2 :164–168, 1944.

[39] F. Lu and E. E. Milios. Optimal spline fitting to planar shape. *Signal Processing*, 37(1) :129–140, 1994.

[40] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 11 :431–441, 1963.

[41] H. Martinsson, A. Bartoli, F. Gaspard, and J-M. Lavest. Handling missing data in the computation of 3d affine transformations. In *5th IAPR International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 90–106, St. Augustine, Florida, USA, 2005.

[42] H. Martinsson, A. Bartoli, F. Gaspard, and J-M. Lavest. Alignement de reconstructions tridimensionnelles affines en présence de données images manquantes. In *Congrés Francophone de Reconnaissance des Formes et Intelligence Artificielle*, Tours, France, 2006.

[43] H. Martinsson, F. Gaspard, A. Bartoli, and J-M. Lavest. Adaptive evolution of 3d curves for quality control. In *IEEE International Symposium on Intellingent Signal Processing*, pages 1–6, Alcalá de Henares, Spain, 2007.

[44] H. Martinsson, F. Gaspard, A. Bartoli, and J-M. Lavest. Energy-based reconstruction of 3d curves for quality control. In *6th IAPR International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 414–428, EZhou, Hubei, China, 2007.

[45] H. Martinsson, F. Gaspard, A. Bartoli, and J-M. Lavest. Reconstruction of 3d curves for quality control. In *15th Scandinavian Conference on Image Analysis*, pages 760–769, Aalborg, Denmark, 2007.

[46] G. McLachlan and T. Krishnan. *The EM algorithm and extensions.* John Wiley & Sons, Inc., 1997.

[47] R.G.N. Meegama and J. C. Rajapakse. NURBS snakes. *Image and Vision Computing*, 21 :551–562, 2003.

[48] S. Menet, P. Saint-Marc, and G. Medioni. B-snakes : Implementation and applications to stereo. In *DARPA*, pages 720–726, 1990.

[49] R. Mohr, B. Boufama, and P. Brand. Accurate projective reconstruction. In *Proceedings of the Second Joint European - US Workshop on Applications of Invariance in Computer Vision*, pages 257 – 276, 1993.

[50] V. Moron. *Mise en correspondance de données 3D avec un modèle CAO : Application à l'inspection automatique.* PhD thesis, École Doctorale Sciences Pour l'Ingénieur de l'INSA de Lyon, 1996.

[51] E. Mouragnon. *Reconstruction 3D et localisation simultanée de caméras mobiles : une approche temps-réel par ajustement de faisceaux local.* PhD thesis, École Doctorale Sciences Pour l'Ingénieur de Clermont-Ferrand, 2007.

[52] L. Piegl and W. Tiller. *The NURBS book*. Monographs in visual communication. Springer Verlag, 2nd edition, 1997.

[53] M. Ribo and M. Brandner. State of the art on vision-based structured light systems for 3d measurements. In *IEEE International Workshop on Robotic and Sensors Environments*, 2005.

[54] J. Rissanen. Modeling by shortest data description. *Automatica*, 14 :465–471, 1978.

[55] C. Sbert and A.F. Solé. Stereo reconstruction of 3d curves. In *15th International Conference on Pattern Recognition*, volume 1, 2000.

[56] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6 :461–464, 1978.

[57] M. Siddiqui and S. Sclaroff. Surface reconstruction from multiple views using rational B-splines and knot insertion. In *First International Symposium on 3D Data Processing Visualization and Transmission*, pages 372–378, 2002.

[58] C. V. Stewart. Bias in robust estimation caused by discontinuities and multiple structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(18) :818–833, August 1997.

[59] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *4th European Conference on Computer Vision*, pages 709–720, 1996.

[60] R. Szeliski and S. B. Kang. Recovering 3d shape and motion from image streams using nonlinear least squares. *Journal of Visual Communication and Image Representation*, 5(1) :10–28, March 1994.

[61] J.-Ph. Tardif, A. Bartoli, M. Trudeau, N. Guilbert, and S. Roy. Algorithms for batch matrix factorization with application to structure-from-motion. In *Conference on Computer Vision and Pattern Recognition*, 2007.

[62] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography : A factorization method. *International Journal of Computer Vision*, 9(2) :137–154, 1992.

[63] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1) :35–61, 2002.

[64] P. H. S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In *ICCV '99 : Proceedings of the International Workshop on Vision Algorithms*, pages 278–294, 1999.

[65] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms : Theory and Practice*, pages 298–372. Springer-Verlag, 1999.

[66] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3D camera tracking. In *3rd IEEE and ACM International Symposium on Mixed and Augmented Reality, Arlington, VA*, pages 48–57, November 2004.

[67] T. Viéville and D. Lingrand. Using specific displacement to analyze motion without calibration. *International Journal of Computer Vision*, 31(1) :5–29, 1999.

[68] T. Viéville, D. Lingrand, and F. Gaspard. Implementing a multi-model estimation method. *International Journal of Computer Vision*, 44(1) :41–64, 2001.

[69] G. Wahba. *Spline Models for Observational Data.* Society for Industrial and Applied Mathematics, 1990.

[70] M.W. Walker, L. Shao, and R.A. Volz. Estimating 3D location parameters using dual number quaternions. *Computer Vision, Graphics and Image Processing : Image Understanding*, 54(3) :358–367, 1991.

[71] Y.J. Xiao and Y.F. Li. Stereo vision based on perspective invariance of NURBS curves. In *IEEE International Conference on Mechatronics and Machine Vision in Practice*, volume 2, pages 51–56, 2001.

[72] C. Xu and J. L. Prince. Snakes, shapes and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3) :359–369, March 1998.

[73] H. Yang, W. Wang, and J. Sun. Control point adjustment for B-spline curve approximation. *Computer-Aided Design*, 36 :639–652, 2004.

[74] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis.* Springer Verlag, 1992.