Reconstruction de scènes à partir d'une seule image

Marta Wilczkowiak

Université Joseph Fourier
U.F.R.
Informatique & Mathématiques Appliquées

Politechnika Warszawska
Wydzial Matematyki
i Nauk Informatycznych

DEA IMAGERIE VISION ROBOTIQUE

Project presented by

Marta Wilczkowiak

3D model acquisition from single image

Prepared in the INRIA Rhône-Alpes

in the MOVI project

Date : JUNE 2000

Supervisors: Dr Edmond Boyer
Prof. Krzysztof Marciniak
Dr Peter Sturm

Contents

Chapter 1

Introduction

The human eye, connected with the human mind, is the base tool that allows us to perceive and interpret the world surrounding us. Our eyes let us not only simply see objects– we can recognize forms, follow movements, estimate distances and velocities. The computer vision is the discipline, which tries to apply those capabilities to computers. This allows to improve and complete our vision and, on the other hand, to eliminate the need of human interaction in certain applications.

The number and importance of applications of computer vision in today's world is difficult to underestimate. Surveillance, robotics, assistance in surgical operations and CAD/CAM systems are only some in the wide range of disciplines which are developed mostly thanks to progress in computer vision. The reasons for applying computer vision are numerous: its flexibility, ability of fast treatment of large amounts of data and, above all, the possibility to express information in a clear and intuitive way.

One branch of computer vision deals with the extraction of threedimensional information from images. The ability of interpretation of 3D data is necessary in some kinds of applications. Consider for example robotics, the problem of obstacle avoidance, or applications for 3D model visualisation.

In this work we are interested in using computer vision to construct threedimensional models of scenes. Reconstruction of geometric models allows us to observe scenes from different points of view. Texture information, which can be extracted from images, permits to obtain photorealistical renderings of models.

There are many solutions to the problem of 3D reconstruction. One approach is a reconstruction of the 3D model with the use of several images of the scene. But we are not always able to obtain two or more images of the scene. Let us consider for example images coming from paintings, archives etc.

On the other hand some special, additional information about the scene can provide strong constraints on the camera parameters and positions of the model points in the 3D space. Especially useful are geometrical dependencies, like parallelism, perpendicularity, incidence of lines or points and lines. Scenes, which provide data of this type, are for example urban or indoor scenes.

The aim of this project is to reconstruct the 3D model of the scene having in disposition only one image. Necessary steps to do so are camera calibration and extraction of the 3D textured information. We will be interested mainly by images of indoor and outdoor architectural scenes. It is easy to observe, that such scenes usually contain some common elements, like different types of cubes, triangle prisms, solids built of sets of parallelograms etc. In this work we will study the different amounts of information provided by projections of such primitives in the image and exploit them for camera calibration and reconstruction of the 3D model of the scene.

1.1 State of the art

The problems of camera calibration and 3D reconstruction have been studied for a long time. Proposed approaches make use, in general, of two or more images, either calibrated or not, which lead to either Euclidean or a projective reconstruction. They are explained for example in Faugeras book [Fau93].

During the last years few approaches to the problem of Euclidean reconstruction from a single image where introduced. Usually the camera is calibrated by exploitation of constraints given by vanishing points of space directions. To the problem of $3D$ reconstruction few different solutions are given which are described below.

B.Caprile et V.Torre [CT90] propose a method of camera calibration based on vanishing points. The projections of two parallel lines in the scene are two lines in the image. Their intersection gives the vanishing point of the

direction. The position of this point is constrained by the intrinsic parameters of the camera. This fact allows to construct equations useful for camera calibration. In this approach the constraints come from the vanishing points of perpendicular directions. The problem in such approaches is the frequently bad numerical conditioning in the vanishing point computation.

Edmond Boyer and Roberto Cipolla [CB98] propose a method for reconstruction of urban environments based on iterative camera calibration. No information a priori about the camera parameters is needed. Calibration is based also on vanishing points. This approach is based on a linear criterion which optimises a three dimensional direction, the dual of a vanishing point in the 3D space. To define paralelism and perpendicularity, cuboids are used. Experiments show, that estimating a space direction is more robust to noise perturbations than estimating an image point. This method can calibrate several images. The reconstruction is provided with the use of few calibrated images.

Fabien Vignon and Edmond Boyer [Vig99] propose a method of modelling of urban scenes based on few images. Their method of camera calibration is based on constraints for camera parameters provided by the images of cuboids. The constraints are extracted from the projection matrix in affine system coordinates attached to the cuboid. This method of calibration applied to parallelepipeds is used in this work. The reconstruction is based on corresponding points in few images.

Peter Sturm and Stephen Maybank [SM99b] present a general algorithm for plane-based calibration, that can deal with arbitrary numbers of views and calibration planes. The algorithm can simultaneously calibrate different views from a camera with variable intrinsic parameters. For some cases they describe the singularities. Experimental results exhibit the singularities and reveal a good performance in non-singular conditions. Also the proposition of reconstruction based on the knowledge of the metric structure of planes is presented.

The same authors in [SM99a] propose a method for interactive 3D reconstruction of piecewise planar objects from a single image. For camera calibration the method explained in [CT90] is used. The reconstruction is based on user-provided coplanarity, perpendicularity and parallelism constraints. Known intrinsic parameters of the camera allow finding projection rays of points. The vanishing line of planes give information about their normal vectors. Unknown are the distances of points and planes from the projection

center. This information, up to scale factor, is obtained by minimization of the sum of squared distances between planes and points. Subsequently, planes with unknown vanishing lines are reconstructed in an iterative manner.

A.Criminisi, I.Reid and A. Zisserman [CRZ99] describe a method of measurements of distances in the 3D scene based on known vanishing line of a reference plane and the vanishing point of the direction perpendicular to the reference plane. A new algebraic representation is developed for measuring distances between planes parallel to the reference plane, computing area and length ratios on planes parallel to the reference plane, determining the camera position. The disadvantage of their sequential reconstruction method is the fact, that errors are not spread uniformly in the model.

An approach for camera calibration based on Parameterised Cuboid Structure is presented in [CSC99] . This method uses a reference parallelepiped in the image to calibrate the camera and create the coordinate system allowing inserting virtual objects into the scene. Some studies on constraints given by known parallelepiped parameters are provided. The camera calibration is made with the quite strong assumption that the aspect ratio of the camera is known.

1.2 Our approach

The aim of this work is to provide methods for 3D scene reconstruction based on a single image. We want to extract information of two different types: geometrical information (position of points, lines, planes) and photometric (texture of surfaces). We are interested mostly in indoor and outdoor architectural scenes. They usually contain geometrical regularities that provide constraints useful for camera calibration and 3D reconstruction. Examples are the perpendicularity, parallelism of lines and planes and various forms of incidence between planes, lines and points. Our special interest was to make exhaustive studies of the constraints provided by primitives, which appear very often in architectural scenes: parallelepipeds and parallelograms. The use of predefined primitives which are common in scenes of interest allows to minimize the work of a user, who introduces the data, while enabling consistant and accurate 3D reconstructions.

The proposition for camera calibration is a logical continuation of the

work of Fabien Vignon and Edmond Boyer [Vig99]. To avoid the use of vanishing points for camera calibration, a method based on constraints provided by the projection matrix from the affine 3D space generated by cuboid structure to the 2D image space is used. In the beginning we studied various kinds of constraints for camera intrinsic parameters provided by projections of corners of reference parallelepipeds. We wanted to represent the parallelepiped in a general way, allowing the optimal exploitation of each of its known parameters. Some work in this area were done in [CSC99], but the study was not exhaustive. Our solution allows to derive one additional constraint from every known parameter. Solving quadratic equations system can do it. The camera calibration tool we developped exploits all the information provided by reference parallelepipeds which leads to constraints of linear type. It gives the possibility of combining the constraints provided by the projections of several reference parallelepipeds. It gives also the possibility of using camera parameters known a priori. For example the assumption that the principal point is situated in the center of the image is very reasonable in general. Also sometimes we know or we can make an assumption concerning the aspect ratio of the camera.

As second step we evaluated a method of reconstruction of the 3D coordinates of points belonging to a connex set of parallelograms. With known camera intrinsic parameters we are able to reconstruct the projection rays of points. To find the depths of those points we evaluated a method introduced in [CSC99] of finding the relative depths of vertices of parallelograms. We wanted to reconstruct many scene points simultaneously to avoid the increasing error in the 3D model, which often occurs in sequential approaches.

1.3 Overview of this report

In the second chapter we give some definitions of projective geometry, camera modelling and calibration, and 3D reconstruction. In the third chapter we introduce the parameterisation of parallelepipeds and present constraints on camera intrinsic parameters provided by known parameters of reference parallelepipeds. We also show experimental results of tests of application based on the presented ideas. In the fourth chapter we present an algorithm for 3D reconstruction based on sets of connex parallelograms, a way of obtaining the correct texture and reconstruction results for real

scenes. In the fifth chapter we present our conclusions and perspectives of future developments of the project.

Chapter 2

Theoretical background

In this chapter we present the theoretical background of this work. At the beginning of this chapter we give notations necessary to read this report. After this we introduce basic definitions concerning projective geometry. Next we briefly describe the principles of camera modelling and calibration. Then we explain the concept of 3D reconstruction and describe the difference between the projective, affine and Euclidean reconstruction. At the end, we briefly present the idea of data normalization, introduced in computer vision by Hartley, which can be used for numerical stabilization of algorithms.

2.1 Notation

Bold letters represents column vectors. We do not distinguish between a point and its algebraical representation. To represent points in 3D space we use capital letters, to represent points in 2D space we use lower case. Corresponding points in the 3D scene and 2D image are represented by the same letter, e.g. $\mathbf{Q}$ represents the point in 3D space and $\mathbf{q}$ represents its projection in the image. The null vector of the space of n dimensions will be represented as $\mathbf{0}_n$.

To note matrices we use capital letters. The identity matrix of dimension n×n is represented by I_n.

Scalars, in particular elements of vectors, are in lower case, for example we can write $\mathbf{Q} = (q_1, q_2, q_3)^\mathsf{T}$.

The vector and dot products of two vectors are represented, respectively,

by $\wedge$ and $\cdot$.

The real space is represented by $\mathcal{R}^n$. The projective space of dimension n is represented by $\mathcal{P}^n$. The projective equality up to a scalar factor is denoted as $\sim$.

2.2 A few notions of projective geometry

2.2.1 Homogeneous coordinates

The basic mathematical framework of our work is projective geometry. The n-dimensional projective space is denoted $\mathcal{P}^n$. Points in $\mathcal{P}^n$ are represented by vectors with $n+1$ coordinates. They are called homogeneous coordinates and are defined up to a scalar factor:

$$[x_1, ..., x_{n+1}]^\mathsf{T} \sim [x_1{}', ..., x_{n+1}{}']^\mathsf{T} \tag{2.1}$$

if, and only if, $\exists \lambda \neq 0$ such that

$$[x_1, ..., x_{n+1}]^\mathsf{T} = \lambda [x_1{}', ..., x_{n+1}{}']^\mathsf{T} \tag{2.2}$$

Linear transformations of a projective space are called *homographies or colineations.*

2.2.2 Point at infinity

The correspondence between point coordinates in $\mathcal{P}^n$ and in $\mathcal{R}^n$ is described as

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \rightarrow \begin{pmatrix} \frac{x_1}{x_4} \\ \frac{x_2}{x_4} \\ \frac{x_3}{x_4} \end{pmatrix} \tag{2.3}$$

This transformation is not possible when $x_4 = 0$. This might be explained by the fact that the projective space consists of the usual affine space and the plane $x_4 = 0$, which is called the plane at infinity.

To ilustrate the concept of points at infinity consider a projective line. It can be represented as set of points of equation:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} + \begin{pmatrix} \lambda \mathbf{d} \\ 0 \end{pmatrix} \sim \begin{pmatrix} \frac{x_1}{\lambda} \\ \frac{x_2}{\lambda} \\ \frac{x_3}{\lambda} \\ \frac{1}{\lambda} \end{pmatrix} + \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} \quad \overrightarrow{\lambda \to \inf} \quad \begin{pmatrix} \mathbf{d} \\ 0 \end{pmatrix} \tag{2.4}$$

Equation(**??**) implies, that all projective lines in direction $\mathbf{d}$ intersect at plane at infinity in the same point $[\mathbf{d}^\mathsf{T}, 0]$ (fig. **??**).

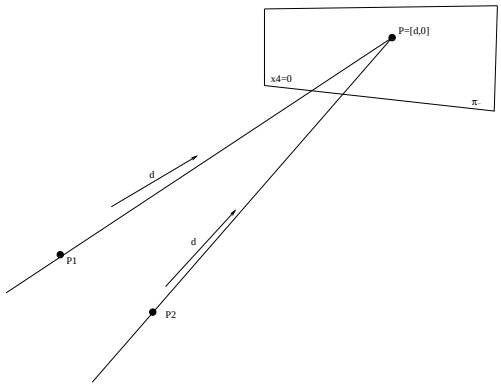


Figure 2.1: The intersection of parallel lines is a point at infinity.

2.2.3 The Absolute Conic

The aim of this section is to introduce the concept of absolute conic, which is useful for camera calibration. At the beginning we give brief definitions of conics and quadrics in projective space, which will allow to define differences between projective, affine and euclidean transformations of the projective space. This will allow us to show the importance of properties of the image of the absolute conic.

A *conic* in $\mathcal{P}^2$ is defined as the curve on the projective plane consisting of points $[x_1, x_2, x_3]^\mathsf{T}$ that satisfy a quadratic equation:

$$\sum_{i,j=1}^{3} a_{ij} x_i x_j = 0 \tag{2.5}$$

A *quadric* of the $\mathcal{P}^3$ space is defined as the surface consisting of points

$[x_1, x_2, x_3, x_4]^\mathsf{T}$that satisfy an equation of the type:

$$\sum_{i,j=1}^{4} a_{ij} x_i x_j = 0 \tag{2.6}$$

The transformations of the projective space, which preserve the plane at infinity are called the *affine transformations*. An important property is the fact that they preserve parallelism of lines and ratios of distances on parallel lines.

Let us define the special conic, consisting of points $[x_1, x_2, x_3, x_4]^\mathsf{T}$ with:

$$\sum_{i=1}^{3} x_i^2 = x_4 = 0 \tag{2.7}$$

This conic is called the *absolute conic*. It consists of complex points at infinity. The transformations of the projective space, which preserve the absolute conic are called the *euclidean transformations*.

The euclidean transformations preserve angles and ratios of lengths of objects. In other words, they can always be decomposed into rotation, translation and a global scaling. Thus the image of the absolute conic does not depend on the position and orientation of the camera in space. It depends only on the intrinsic parameters of the camera. That is why the concept of the absolute conic is basic for camera calibration.

2.3 Camera modelling

In this section we describe some basic concepts for camera modelling. At the beginning we describe the camera model used in this work. In the second part we briefly present the transformations that are applied to points in 3D space to obtain their projections in the image.

The camera model we use is the pinhole model. It is shown on the figure ??.

The pinhole camera model of the camera is based on the projective transformation of the 3D space to the plane called the *image plane*. The projection center $\mathbf{C}$ is also called the *optical center*. The distance between the optic center and image plane is called the *focal distance*. The image $\mathbf{x}$ of the point $\mathbf{X}$

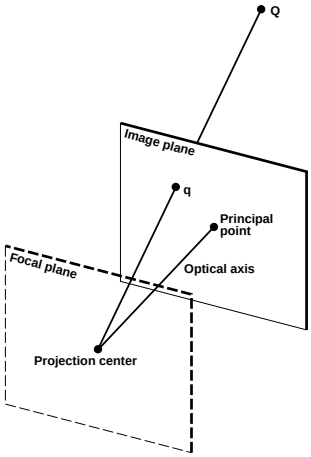


Figure 2.2: The pinhole camera model.

in space is the intersection of the line **XC** and the image plane. This line is called the *projection ray*. The *optic axis* is the line passing through the projection center and that is perpendicular to the image plane. Its intersection with the image plane is called the *principal point*. The *focal plane* is the plane containing the projection center and parallel to the image plane. It is easy to observe, that projections of points in the focal plane are points at infinity of the image plane.

For the mathematical description of the pinhole model, we consider normally four different coordinates systems: a scene system, a camera system, a system in the image plane and a system related to the pixel geometry of the image plane. The relation between those systems is shown on the figure **??**:

The transformation between the point coordinates in the 3D scene and coordinates of its projection in the image is composed of the following transformations:

$$\begin{pmatrix} wu \\ wv \\ w \end{pmatrix} = {}^{i}_{r}C \; {}^{r}_{c}S \; {}^{c}_{s}T \begin{pmatrix} x \\ y \\ z \\ t = 1 \end{pmatrix} \tag{2.8}$$

T is the transformation from the scene coordinate frame to the camera coordinate fram. It is a euclidean transformation, so it can be decomposed into a rotation R and a translation $\mathbf{t}$. S represents the perspective projection of

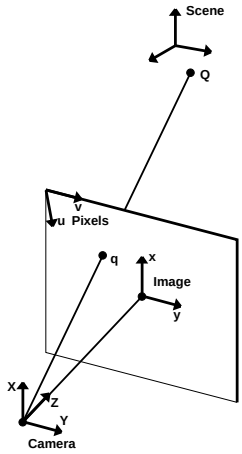


Figure 2.3: The pinhole camera model coordinate systems.

points in the coordinates system connected with the camera to the retinal plane. C is the affine transformation from the coordinate system connected with the image plane to the system connected with pixels. The composition of those transformations is usually represented by a *projection matrix* $\mathsf{P}_{3\times 4}$. As the result we have:

$$\mathbf{x} \sim \mathsf{P}_{3\times 4}\mathbf{X} \tag{2.9}$$

Matrix $\mathsf{P}$is defined up the scale factor, so it has $3 \times 4 - 1 = 11$ degrees of freedom.

The projection matrix may be decomposed as follows in order to exhibit physical properties of the camera:

$$\mathsf{P} \sim \mathsf{K}(\quad \mathsf{R} \quad | \ \mathbf{t}) \tag{2.10}$$

Where R_{3x3} is the rotation matrix (an orthogonal matrix) and $\mathbf{t}$ a "translation vector" of size 3. The R and $\mathbf{t}$ have six degrees of freedom and they are called *extrinsic parameters* of the camera. They represent the orientation and position of the camera with respect to the 3D scene. The matrix K is the upper triangular and has thus five degrees of freedom(it is defined up to scale). They correspond to the *intrinsic parameters* of the camera. Once the intrinsic parameters of the camera are known, we say that camera is calibrated.

The matrix K of intrinsic parameters of the camera can be represented

as follows:

$$\mathsf{K} = \begin{pmatrix} \tau f & s & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad .$$

Its elements have the following meanings:

(u_0, v_0) - the position of the principal points in the image
τ - the aspect ratio i.e. the ratio between horizontal and vertical dimensions of a pix
s - the skew parameter which depends on the angle between axes. As normally pixel
f - the focal distance multiplied by height of a pixel in units of a retinal frame.

2.4 Camera calibration

As we mentioned at the end of the previous section, by calibration of the camera we understand the extraction of its intrinsic parameters.

There are two main approaches to the problem of the camera calibration:

- If we know positions of points in space– such as points on a calibration pattern–we can calibrate the camera using coordinates of their projections in one or more images.Equation (**??**) gives us two constraints on the elements of the projection matrix for every known correspondence between a 3D point and its image. That implies that if we know positions in the euclidean space of at least five points and one direction, we can obtain the projection matrix. Then, using the fact, that K is upper triangular and R is orthogonal, we can extract the intrinsic parameters of the camera by Cholesky decomposition.

- Other methods of camera calibration use constraints provided by known geometrical dependencies, like perpendicularity and parallelism of directions in the scene. There exist several approaches to this problem, but they lead usually to similar constraints on the intrinsic parameters, i.e. to equations based on the image of the absolute conic. Its link to calibration and metric scene structure was described in section **??**.

  To demonstrate this type of constraints briefly describe the method of camera calibration based on vanishing points as introduced in [CT90].

  The vanishing point of a direction is the image of the point of intersection of the parallel lines in the space, which is a point on the plane at

infinity. It is shown on the figure **??**.

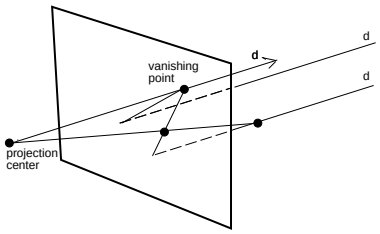


Figure 2.4: The vanishing point.

The main idea of the algorithm is based on the fact, that the scalar product of two perpendicular directions is equal to 0. Let us take two ideal points, i.e. points at infinity, $[\mathbf{V}_1, 0]^\mathsf{T}$ and $[\mathbf{V}_2, 0]^\mathsf{T}$. From the projections equations $\mathbf{v}_i \sim \mathsf{K}\mathbf{V}_i$ we obtain the correspondence

$$\mathbf{V}_i \sim \mathsf{K}^{-1}\mathbf{v}_i \tag{2.11}$$

Considering this and the fact that ${\mathbf{V}_1}^\mathsf{T} \cdot \mathbf{V}_2 = 0$ we have:

$$\mathbf{v}_1\mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}\mathbf{v}_2{=}0 \tag{2.12}$$

where the $\mathsf{K}^{-\mathsf{T}}$ is the transposition of invertion of K. The matrix $\mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}$ is the image of the absolute conic.

Equation(**??**) implies that every pair of vanishing points of perpendicular directions gives one constraint on intrinsic parameters of the camera.

This shows the utility for calibration of images of cuboids. One cuboid in general position (**??**) brings three constraints on the intrinsic parameters. In many practical situations some *a priori* information on the intrinsic parameters is given; so the constraints provided by a cuboid are usually sufficient to calibrate the camera.

2.5 3D Reconstruction

By the reconstruction of the 3D model of the scene we understand extraction of object positions in space. Depending on data at hand, we can

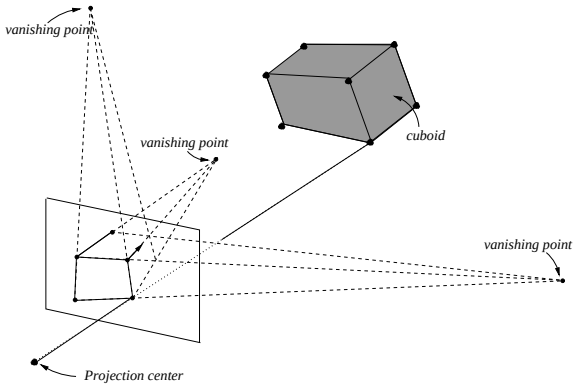


Figure 2.5: Vanishing points implied by the cuboid.

obtain different types of reconstruction. Usually we are interested in reconstruction, which preserve the angles and at least ratios of lengths. This kind of reconstruction is not possible if we do not have the camera calibrated.

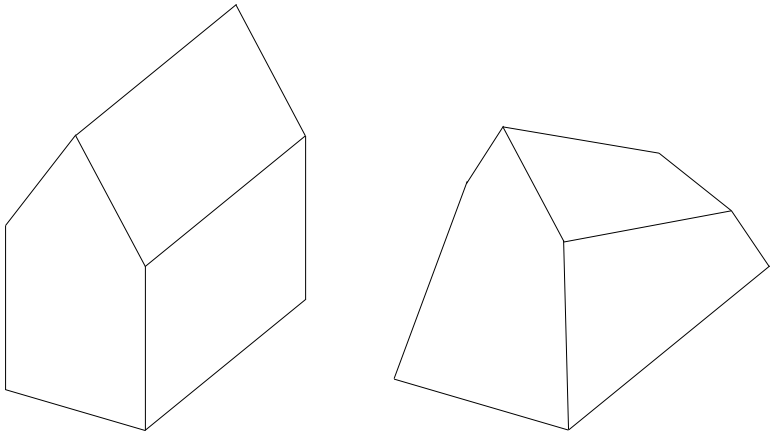
2.5.1 3D Projective reconstruction

In the case of the uncalibrated camera we are only able to obtain a so-called projective reconstruction. It brings some 3-dimensional information, allowing for example to recognize the type of the object(see figure **??**). It was shown by Faugeras in [Fau92] and Hartley in [HGC92] that it is possible to obtain the projective calibration of a set of cameras without any additional information about the scene structure.

2.5.2 3D Affine and euclidean reconstruction

By affine reconstruction we understand a reconstruction, which preserves the parallelism and the lengths ratios on parallel directions. To obtain this type of reconstruction we have to find the transformation from the projective space to the affine space. To find this transformation we have to identify the plane at infinity. Three points in general positions belonging to the plane define it. They can be obtained by use of the vanishing points (fig. **??**) of three pairs of parallel lines. It was shown that using this concept it is quite easy to obtain the affine reconstruction.

In order to obtain the most interesting, euclidean reconstruction or metric



(a) The euclidean model (up to the scale vector).

(b) The projective model. Coplanarity and general form are preserved.

Figure 2.6: Two threedimensional models of the house.

reconstruction, we have to extract the intrinsic parameters of the camera. For this, some additional information has to be given. One useful constraint is the constraint of perpendicularity ([CT90],[CB98]). Known angles, some knowledge about distance ratios, can also bring useful constraints, as shown in chapter **??**

The euclidean reconstruction, without any metric information about the scene, can be obtained only up to a scalar factor. This is satisfactory in most of applications. But it is sufficient to know a single length to fix the scalar factor and obtain an exact 3D reconstruction of the scene.

2.6 Data normalisation

One problem that we face almost every time when we apply ideas to real systems is the bad conditioning of the data. Let us consider image points, which are the base of our computations. They are usually of the type $\mathbf{x}=[u, v, 1]^{\mathsf{T}}$, where u, v are of the order $10^2, 10^3$. The type of constraints that are usually used (described in **??**) implies, that often some coefficients are

of the order between 1 and 10^6. The other thing is that if pixels are not centered, their coordinates are of very different orders.This might causes a bad conditioning of equation system and lead to very unstable solutions.

R. Hartley in [Har95] proposed a normalization procedure for points such as to obtain coordinates $\mathbf{x}'=[u', v', 1]^{\mathsf{T}}$, where u', v' are of the order of 1. It is obtained by a change of the image coordinate system in a way that guarantees that all points of interest are lying in the unit circle. The first step is the translation of the center of the coordinate system to the gravity center of the set of points. The second step is a scaling of coordinates to obtain lengths of vectors $\mathbf{x}_i^{'} \sim 1$.This idea is illustrated by fig.**??**.

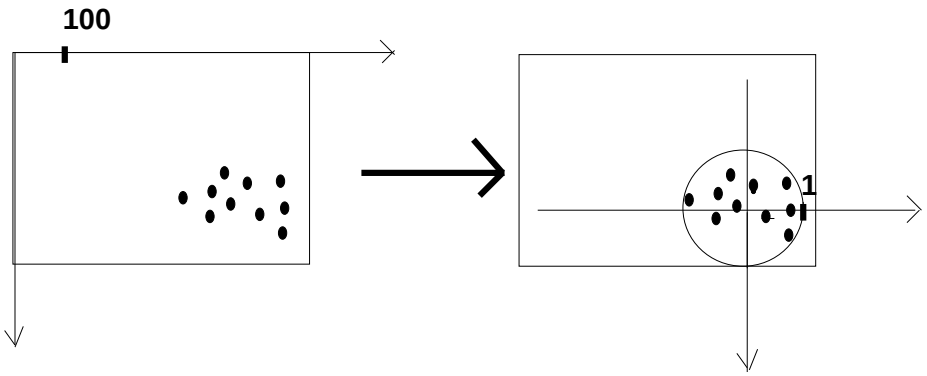


Figure 2.7: The Hartley normalisation.

Such transformations improve the conditionning and lead to better results and more robust alghorithms [Har95].

Chapter 3

Camera calibration using parallelepipeds

In this chapter we would like to introduce a calibration method based on parallelepipeds.

We especially are interested in urban scenes. The base primitive existing in man–made environments is the cuboid (Fig. ??).

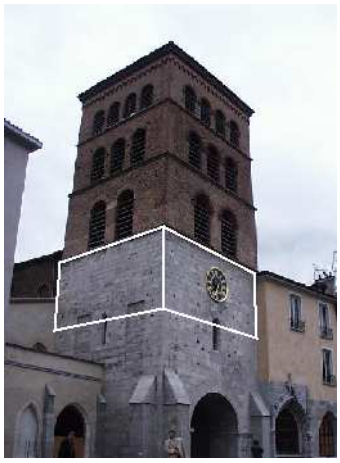
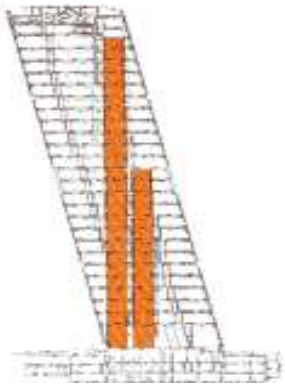


Figure 3.1: Example of cuboid definition for camera calibration

The utility of cuboid structures in defining urban scene structure is obvious. But it is also clear, that we can not define the entire scene only by means of cuboids. On the other side the invention of architects is usually limited

by constraints of space and facility of construction, which leads to using of the structure more interesting than cuboid, but still quite easy to construct-the parallelepiped structure. We can see the examples of such buildings on Figures ??, ??.

(a) Torres Kio, Madrid.



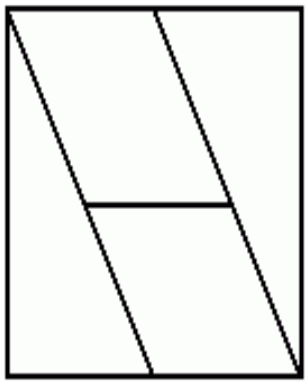
(b) Two towers lean towards each other at an angle of 15^o.

Figure 3.2: Example of parallelepiped structure in urbane scene.

The second reason speaking for applying parallelepipeds is the fact, that they extend the class of object which can be used for calibration, so we can obtain more constraints. In the presence of the noisy data it can numerically stabilize results. The Figure ?? shows how we can define additional parallelepiped structure, which is not explicitly shown on the image.

Our method exploits the constraints on intrinsic parameters provided by images of vertices of parallelepipeds in the image and known parameters of the parallelepipeds. In order to express those constraints, we had to choose a parameterization of the parallelepipeds. This is described in the first part of the chapter. Then we show our method for computing an intermediate projection matrix from the affine coordinate system related to parallelepiped structure. Next we present constraints on the camera intrinsic parameters provided by columns of this intermediate matrix. Then we describe our way of using the prior knowledge on intrinsic parameters and discuss which parameters can be calibrated in degenerated cases. Finally, we describe our programming approach and results of test.

(a) John Hancock Tower, Boston.



(b) The shape of the John Hancock tower is rhomboid.

Figure 3.3: Example of parallelepiped structure in urbane scene.

3.1 Parallelepiped parameterization

Our method of camera calibration is based on matches between projections of the parallelepiped's corners in the image and their coordinates in the affine space related to the primitive.

Let us define:

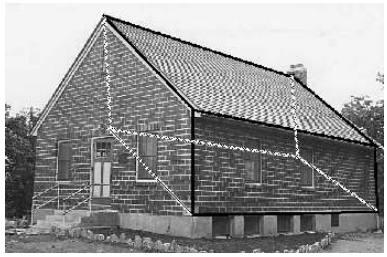
- $\theta_1 = \angle\mathbf{XY}, \theta_2 = \angle\mathbf{YZ}, \theta_3 = \angle\mathbf{ZX}$ –angles between the edges of parallelepiped.
- $\lambda_1, \lambda_2, \lambda_3$ –lengths of edges

To define constraints on the intrinsic parameters, we introduced the parameterization of parallelepiped shown in Figure **??**.

The basis change from affine to euclidean space can be described as:

$$^{E}\mathbf{X}{=}\Lambda_4 \ {}^{A}\mathbf{X} \tag{3.1}$$

(a) The cube used for calibration.



(b) An additional parallelepiped structure.

Figure 3.4: The use of parallelepipeds for additional constraints.

where Λ_4 is the matrix of change of the base and can be expressed as

$$\Lambda_4 = \begin{pmatrix} x_1 & x_2 & x_3 & 0 \\ 0 & y_2 & y_3 & 0 \\ 0 & 0 & z_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.2}$$

where

$$x_1 = \lambda_1 \tag{3.3}$$
$$x_2 = \lambda_2 \cos\theta_1 \tag{3.4}$$
$$y_2 = \lambda_2 \sin\theta_1 \tag{3.5}$$
$$x_3 = \lambda_3 \cos\theta_3 \tag{3.6}$$
$$y_3 = \lambda_3 \frac{\cos\theta_2 - \cos\theta_3 \cos\theta_1}{\sin\theta_1} \tag{3.7}$$
$$z_3 = \pm\lambda_3 \sqrt{1 - (\cos\theta_3)^2 - \left(\frac{\cos\theta_2 - \cos\theta_3 \cos\theta_1}{\sin\theta_1}\right)^2} \tag{3.8}$$

3.2 Computation of the intermediate projection matrix

Let us fix now the coordinates of a parallelepiped's vertices in an affine system related to the primitive, for example as shown on the Figure **??**.

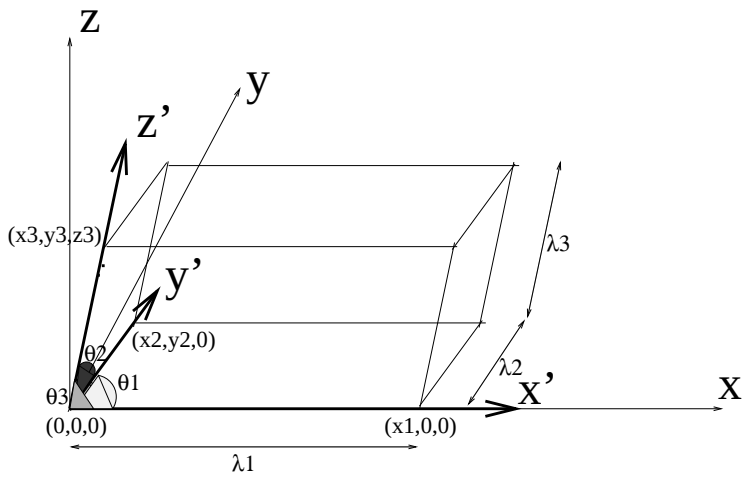


Figure 3.5: The affine coordinate system related to the parallelepiped.

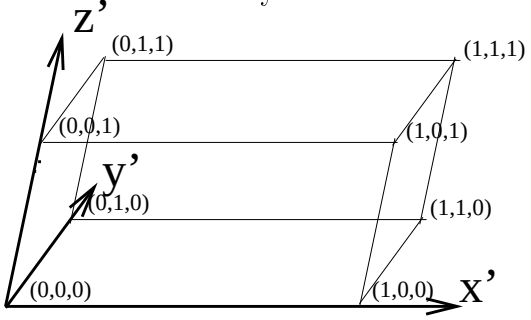


Figure 3.6: The fixed coordinates of vertices in the affine coordinate system related to the parallelepiped.

Let $\mathbf{Q}_i$ be the coordinates of vertex i in the coordinate system of the parallelepiped and $\mathbf{q}_i$ its projection in the image. From the previous section we know that the coordinates of the point $\mathbf{Q}_i$ in the euclidean coordinate system are expressed by $\Lambda_4\mathbf{Q}_i$. Let P be the projection matrix from euclidean space to the image. We have the following dependence:

$$P\Lambda_4\mathbf{Q}_i = \mathbf{q}_i \tag{3.9}$$

Every dependence of the type **??**(–i.e. every known projection of one of the parallelepiped's vertices –) gives us the following constraints on the matrix $\mathsf{M} = \mathsf{P}\Lambda_4$:

$$\begin{pmatrix} wu \\ wv \\ w \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} = 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ t = 1 \end{pmatrix} \tag{3.10}$$

This allows the construction of two linear equations from every dependence:

$$\begin{array}{c} m_{31}xu + m_{32}yu + m_{33}zu + u = m_{11}x + m_{12}y + m_{13}z + m_{14} \\ m_{31}xv + m_{32}yv + m_{33}zv + v = m_{21}x + m_{22}y + m_{23}z + m_{24} \end{array} \tag{3.11}$$

which leads to the following equation system of the type: $\mathsf{A}\mathbf{X} = \mathsf{B}$

$$\begin{pmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 & -z_1u_1 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -x_1v_1 & -y_1v_1 & -z_1v_1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix} \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} = 1 \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ \vdots \end{pmatrix} \tag{3.12}$$

The dimension of the solutions' space of the homogenous system $\mathsf{AX} = \mathsf{B}$ depends on the rank of A.

- if rk $A = 11$ one unique solution, up to a scalar factor, is obtained
- si rk $A < 11$ an infinite family of solutions is obtained.

This implies that we need at least six points in general position to obtain the solution. In theory we need only five points and one direction (a direction provides one constraint) to obtain the matrix. In practice we can usually identify at least six projections of the primitive's vertices. On the other hand, to improve the results, we usually use overconstrained systems. We find the solution of such a system by minimization of the expression $\|\mathsf{AX} - \mathsf{B}\|$ in the sense of the least squares using the SVD method (described for example in [PTVn92].

3.3 Constraints on the intrinsic parameters

In this section we will describe the constraints for the camera intrinsic parameters provided by matrix $\mathsf{P}\Lambda_4$ introduced in the previous section.

Using the equation (**??**) we can write

$$\mathsf{P}\Lambda_4 \sim \mathsf{K}(\quad \mathsf{R} \quad | \ \mathbf{t})\Lambda_4 \tag{3.13}$$

Let us consider the matrix $\mathsf{X}_{3\times 3} = \mathsf{KR}\Lambda$, where

$$\Lambda = \begin{pmatrix} x_1 & x_2 & x_3 \\ 0 & y_2 & y_3 \\ 0 & 0 & z_3 \end{pmatrix} \tag{3.14}$$

Let us evaluate the expression $X^{\mathsf{T}}\mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}X$ using the fact that R is the orthogonal matrix. We obtain:

$$X^{\mathsf{T}}\mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}X = (\mathsf{KR}\Lambda)^{\mathsf{T}}\mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}\mathsf{KR}\Lambda = \Lambda^{\mathsf{T}}\mathsf{R}^{\mathsf{T}}\mathsf{K}^{\mathsf{T}}\mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}\mathsf{KR}\Lambda = \Lambda^{\mathsf{T}}\Lambda \tag{3.15}$$

The matrices $\Lambda^{\mathsf{T}}\Lambda$ and $\omega \sim \mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}$ are of the following form:

$$\Lambda^{\mathsf{T}}\Lambda \sim \begin{pmatrix} \lambda_1^2 & \lambda_1\lambda_2\cos\theta_1 & \lambda_1\lambda_3\cos\theta_3 \\ \lambda_1\lambda_2\cos\theta_1 & \lambda_2^2 & \lambda_2\lambda_3\cos\theta_2 \\ \lambda_1\lambda_3\cos\theta_3 & \lambda_2\lambda_3\cos\theta_2 & \lambda_3^2 \end{pmatrix} \tag{3.16}$$

$$\omega \sim \begin{pmatrix} 1 & 0 & -u_0 \\ 0 & \tau^2 & -\tau^2 v_0 \\ -u_0 & -\tau^2 v_0 & \tau^2 f^2 + u_0^2 + \tau^2 v_0^2 \end{pmatrix} . \tag{3.17}$$

We are looking for elements of ω only up to a scalar factor, so we can fix one of its elements, for example ω_{11}. Considering equations (??), (??) and (??) we obtain:

$$\mathbf{X}^{\mathsf{T}} \begin{pmatrix} \omega_{11}=1 & 0 & \omega_{13} \\ 0 & \omega_{22} & \omega_{23} \\ \omega_{13} & \omega_{23} & \omega_{33} \end{pmatrix} \mathbf{X} \sim \begin{pmatrix} \lambda_1^2 & \lambda_1\lambda_2\cos\theta_1 & \lambda_1\lambda_3\cos\theta_3 \\ \lambda_1\lambda_2\cos\theta_1 & \lambda_2^2 & \lambda_2\lambda_3\cos\theta_2 \\ \lambda_1\lambda_3\cos\theta_3 & \lambda_2\lambda_3\cos\theta_2 & \lambda_3^2 \end{pmatrix} \tag{3.18}$$

Equation (??) allows to make some observations about what kind of constraints on intrinsic parameters we can obtain depending on the kind of information about parallelepiped parameters. Considering that $\Lambda^{\mathsf{T}}\Lambda$ is known up to a scalar factor, we have:

- One known angle $\Theta_{ij} = 90^o$ gives one linear constraint on elements of ω:

$$\mathbf{X}_i{}^{\mathsf{T}}\omega\mathbf{X}_j = 0 \tag{3.19}$$

- One known angle Θ_{ij} gives one nonlinear constraint on elements of ω:

$$\cos^2\Theta_{ij} = \frac{(\mathbf{X}_i{}^{\mathsf{T}}\omega\mathbf{X}_j)^2}{(\mathbf{X}_i{}^{\mathsf{T}}\omega\mathbf{X}_i)(\mathbf{X}_j{}^{\mathsf{T}}\omega\mathbf{X}_j)} \tag{3.20}$$

- One known length ratio $r_{ij} = \frac{\lambda_i}{\lambda_j}$ gives one linear constraint on ω elements:

$$\mathbf{X}_i{}^{\mathsf{T}}\omega\mathbf{X}_i - (r_{ij})^2 \cdot \mathbf{X}_j{}^{\mathsf{T}}\omega\mathbf{X}_j = 0 \tag{3.21}$$

- Generally, if we know some parameters of the parallelepiped we obtain then up to five constraints on the elements of ω:

$$\mathbf{X}_i{}^{\mathsf{T}}\omega\mathbf{X}_j - \frac{(\Lambda^{\mathsf{T}}\Lambda)_{ij}}{(\Lambda^{\mathsf{T}}\Lambda)_{kl}}\mathbf{X}_k{}^{\mathsf{T}}\omega\mathbf{X}_l = 0 \tag{3.22}$$

If more parallelepipeds are available, we just include new equations into an equation system. At the beginning we add only constraints of the linear type which leads to an equation system of type $\mathsf{A}\mathbf{X} = \mathbf{0}$. Having at least four equations we can obtain a unique solution for the vector of unknowns $[\omega_{13}, \omega_{22}, \omega_{23}, \omega_{33}]^{\mathsf{T}}$ using the SVD method. If the linear system is under-constrained we can use non-linear constraints to limit the space of solutions obtained by applying the SVD method for under constrained equation system.

Once the elements of the matrix ω are known the intrinsic camera parameters can be extracted via [SM99b]:

$$\tau^2 = \frac{\omega_{22}}{\omega_{11}} \tag{3.23}$$

$$u_0 = -\frac{\omega_{13}}{\omega_{11}} \tag{3.24}$$

$$v_0 = -\frac{\omega_{23}}{\omega_{22}} \tag{3.25}$$

$$f^2 = \frac{\omega_{11}\omega_{22}\omega_{33} - \omega_{22}\omega_{13}^2 - \omega_{11}\omega_{23}^2}{\omega_{11}\omega_{22}^2} \tag{3.26}$$

3.4 Prior knowledge about camera parameters, degenerated cases

Very often we know or we can make assumptions concerning some intrinsic parameters of the camera. For most cameras, for example, the principal point lies in the center of the image. Information about the aspect ratio can usually be found in a camera's technical documentation, or can sometimes be assumed to be unit. The focal distance is the only parameter that we always have to calibrate in practice.

On the other hand the position of the calibrating primitive does not always allow to extract all the intrinsic parameters. It is easy to show both algebraically and geometrically that the following degenerated cases exist:

- If one face of the primitive is parallel to the image plane and its lengths are unknown then edges of this face give no constraints on the principal point, focal distance and skew factor(Eq.?? is always true).
- The direction perpendicular to the image plane allows to localize the principal point, but does not constrain other intrinsic parameters.

Those degenerated cases are illustrated by Figure ??

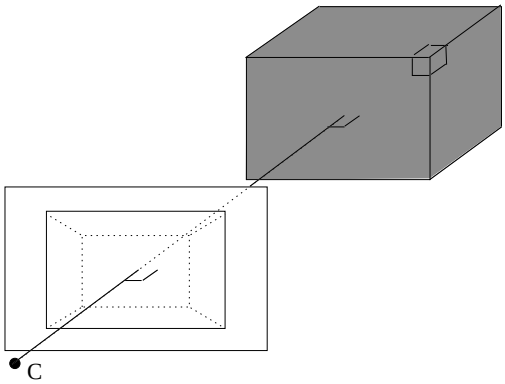


Figure 3.7: A degenerated case. Projections of lines lying in the plane parallel to the image plane do not constrain the intrinsic parameters if lengths are unknown. The projection of the perpendicular to the image plane direction determines the principal point.

There are two ways to use the prior knowledge about the intrinsic parameters.

- 0ne method is to translate the image coordinate system to the principal point and to scale its axes such that in further calculus $u_0 = 0, v_0 = 0, \tau = 1$.
- The second solution is to add to the main equation system, depending on which parameters are known, the following equations:

$$\omega_{13} = -u_0 \tag{3.27}$$
$$\omega_{22} = \tau^2 \tag{3.28}$$
$$v_0\omega_{22} + \omega_{23} = 0 \tag{3.29}$$

3.5 Experimental results

To verify robustness of our methods we made few tests. At the beginning we tested our calibration method on a series of synthetic data. In the second stage we made test based on photo of the calibration pattern. At the end we tested our application on real data. In this section we describe our results.

3.5.1 Synthetic data

To obtain general deductions concerning our method we applied tests on synthetic data. We applied the same test to two different primitives:

- First test was based on parallelepiped with angles $(30^0, 45^0, 60^0)$ and lengths given.
- Second test was based on parallelepiped with angles $(60^0, 90^0, 90^0)$ and unknown lengths.

Our application have few types of parameters. First type are image coordinates of primitives projections. The second type are known *a priori* intrinsic parameters like position of a principal point and a skew ratio. In order to verify the sensitivity of our method to those parameters we applied tests of the method with noisy data.

On the other side, we wanted to verify the behavior of the method in degenerated cases. In order to do it, we tested results of calibration based on parallelepiped rotated around the vertical axis parallel to the image plane.

The calibration results given above were obtained as a median from a series of random test. Each test was provided with and without data normalization in the precomputation stage. We assumed that image size is 512×512 and the principal point lies in the center of the image. We formulated several deductions:

- The general observation is that the normalization smoothes results, but in the case of our method the improvement of results is not very significant. This can be seen on Figures ?? and ??. They present results of calibration with and without normalization for both tested primitives rotated around the vertical axis. We can see that positions close to degenerated cases does not allow to calibrate a camera with or without normalization. Also results of calibration are not very different. Similar deductions were made in tests of sensitivity to noise on program parameters.

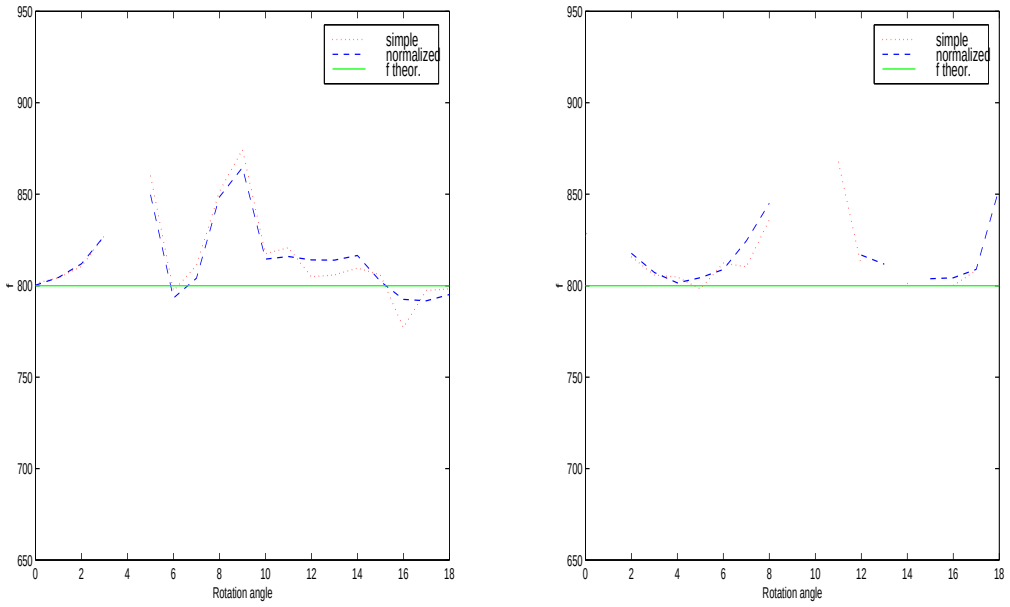
- Knowledge about lengths of parallelepiped, thanks to additional equations to the calibration system, improves obtained results in tests on sensitivity to noise on known a priori intrinsic camera parameters and test on rotated primitives.

- Known lengths of edges allows to calibrate also in otherwise degenerated cases (described in section ??). Figures ?? and ?? show results of calibration of a parallelepiped, which is rotated around the vertical axis parallel to the image plane. The random noise of two pixels was applied to primitives vertices image coordinates. In case of parallelepiped, which lengths are not known, every time its face is parallel to the image plane, calibration is not possible. In case of known edges in such cases we see only increasing error.

- In the second test we applied random noise on image coordinations of primitives vertices projections.

  Calibration in case of primitive whose edges lengths are given is possible up to 3 pixels noise (Figure ??). In the second test results are reasonable up to six pixels noise (Figure ??). It can be explained by the fact, that in case of the first primitive (with all parameters given) we obtain more equations based on noised data than in the second test. It may cause worst results.

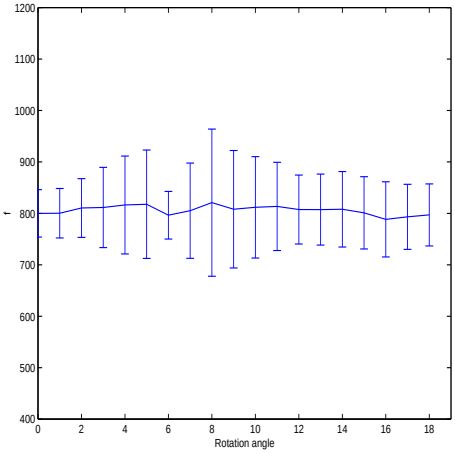
- The third test (Figures ?? and ??) was applied to verify the influence of the error on position of a principal point in the image. We tested behavior of the method with distance between (u_0, v_0) real and given as a program parameter form 0 to 200 pixels. Experiments on both primitives show, that the method is not very sensitive for not exact position of a principal point. Errors on position of the principal point



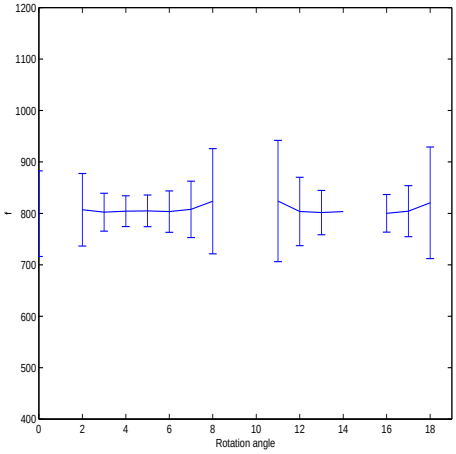
(a) Test 1, rotation

(b) Test 2, rotation

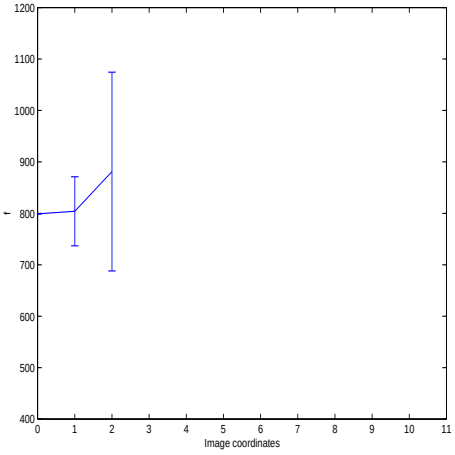
Figure 3.8: Test of calibration based on parallelepiped roted around axis parallel to the image plane for primitive with known lengths(a) and unknown lengths (b). In both cases degenerated cases are seen, but in case (a) calibration is possible. Normalization of data smooths results.



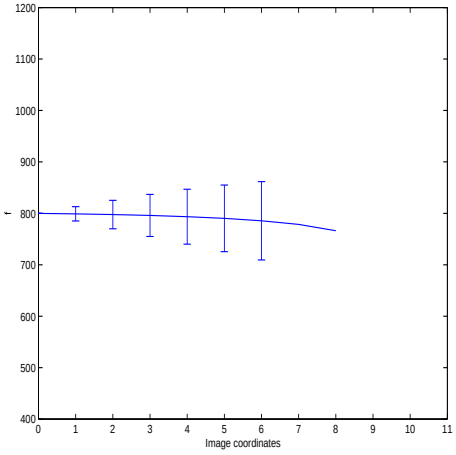
(a) Test 1, rotation



(b) Test 2, rotation

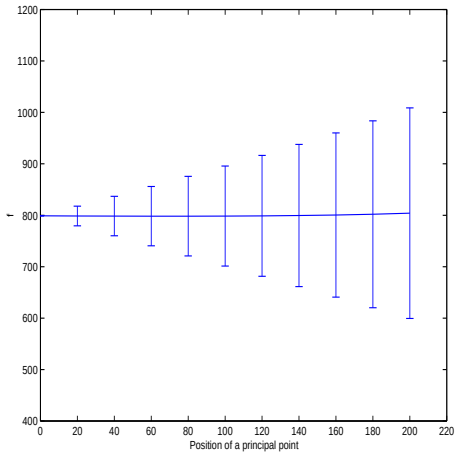


(c) Test 1, noised image coordinates

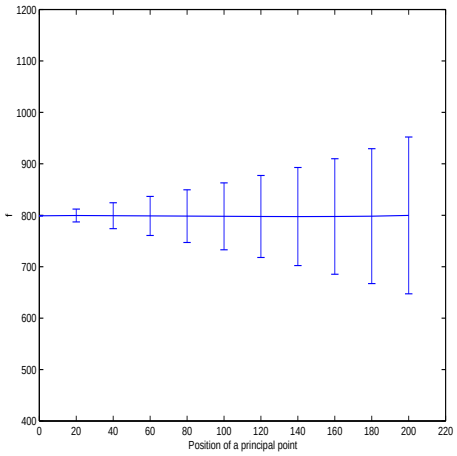


(d) Test 2, noised image coordinates

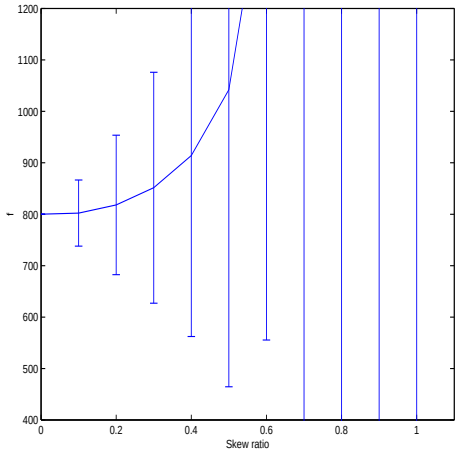
Figure 3.9: Graphs of averages and standard deviations from tests on synthetic data.(a)- test of calibration based on primitive with angle between face and image plane between $0 - 180$, edges lengths are known; (b)- test of calibration based on primitive with angle between face and image plane between $0 - 180$, edges lengths are unknown;(c)- test with error on image coordinates from $0 - 10$ pixels, edges lengths are known;(d)- test with error on image coordinates from $0 - 10$ pixels, edges lengths are unknown;



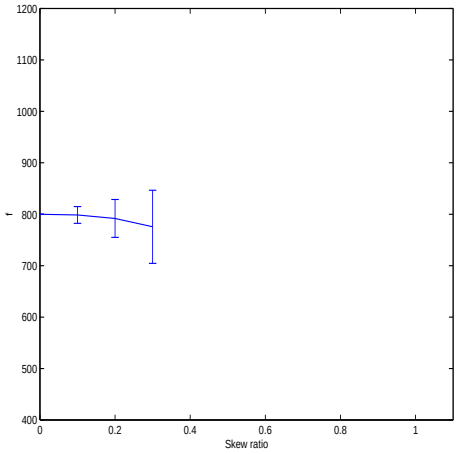
(a) Test 1, noised position (u0,v0)



(b) Test 2, noised position (u0,v0)



(c) Test 1, noised skew ratio



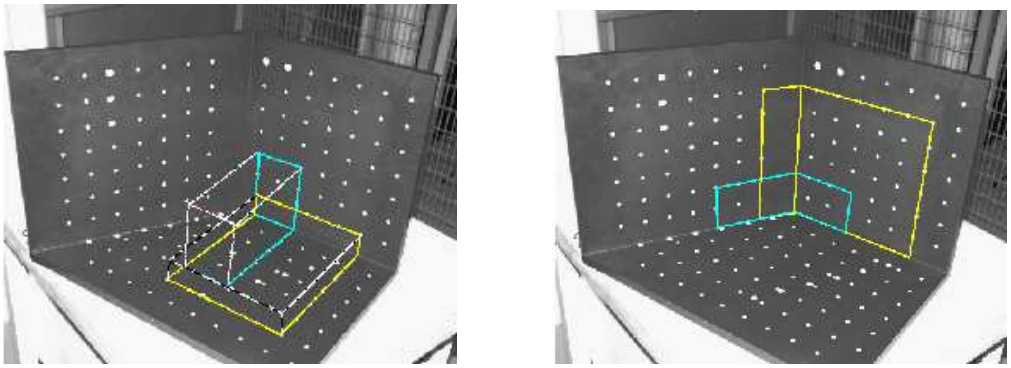
(d) Test 1, noised skew ratio

Figure 3.10: Graphs of averages and standard deviations from tests on synthetic data.(a)- test with error on position of principal point, edges lengths are known; (b)- test with error on position of principal point, edges lengths are unknown;(c)- test with error on skew ratio, edges lengths are known ;(d)- test with error on skew ratio, edges lengths are unknown;

until 50% of the image size gives error on calibrated parameters less then 20%. This result shows, that assumption that the principal point lies in the center of the image is usually very reasonable and even if it is false, calibration should give quite good results.

- Prior knowledge about the aspect ratio improves the calibration. On the other hand if the value given is not exact, it spoils results. It is shown on Figures ?? and ??.

3.5.2 Calibration pattern image



(a) Family of cuboids.

(b) Family of parallelepipeds.

Figure 3.11: Experiment with calibration pattern.

For the second test we used the photo of a calibration pattern. We examined two families of primitives: cuboids(Fig. ??) and parallelepipeds (Fig. ??) with different size of edges. We applied also tests to compare results for every family with known and unknown lengths of edges .In all cases we assumed that the principal point lies in the center of the image. Thanks to this test we made more deductions about dependence of calibration results on orientation and size of the primitive. Real values of intrinsic parameters where computed by standard method using known positions of control points of the calibration pattern in the space.

- In tables ?? and ?? we give exemplary calibration results for primitives which have all three edges at similar length. Generally the bigger the

size of primitive, the better results of calibration we obtain. On the other hand we observe, that primitives too big, so with projections of vertices close to borders of the image, give worst results than primitives well centered. We give exemplary results for primitives with known and unknown lengths of edges.

lenght $\sim$	1	4	5	6
cuboids	49.5	5.6	6.2	3.7
cuboids+lengths	47.4	3.2	3.6	1.6
parallelepipedes	-	3.0	2.5	0.1
parallelepipedes +lengths	-	2.7	2.4	0.4

Table 3.1: Relative errors for f in %

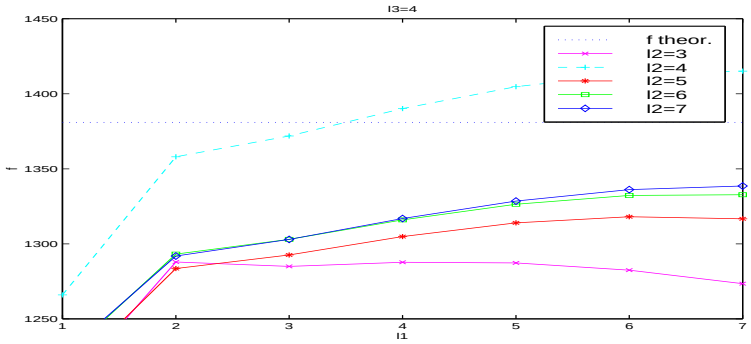
lenght $\sim$	1	4	5	6
cuboids	14.6	3.7	2.9	2.8
cuboids+lengths	9.7	2.2	2.8	2.0
parallelepipedes	-	0.0	2.4	2.2
parallelepipedes+lengths	-	1.1	1.1	0.0

Table 3.2: Relative errors for τ in %

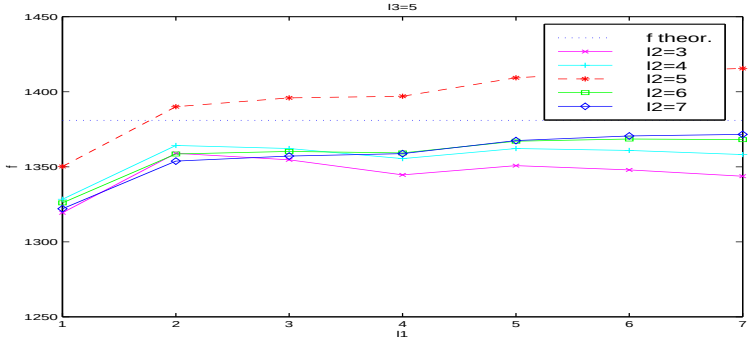
- Calibration is influenced by existence of squared faces. Figures **??** show results. Three charts are made for three different values of the length of the edge (l_3) of the cuboid. On the **x** axis are values of length l_2. The **y** axis presents results of focal computation for different values of l_2. We can observe, that the best results are not for the biggest value of l_2, but for the value closest to the length l_3.

3.5.3 Real scenes

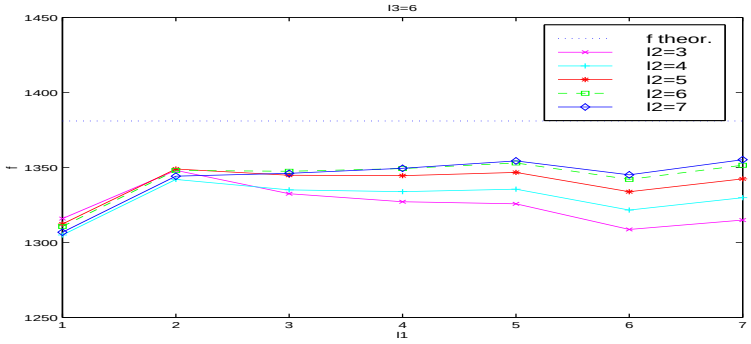
The most interesting in our work were tests on images of real scenes. In this case calibration results are not really the most significant. In practice we do not know intrinsic parameters of a camera, with which the image was taken. It means that we can verify the calibration only by studying properties



(a) $l_3 = 3$



(b) $l_3 = 4$



(c) $l_3 = 5$

Figure 3.12: More important then size of edges is the rectangular shape of the primitive.

of the obtained model. This will be done in section ?? of the next chapter. We will describe then scenes, which had been reconstructed and problems which occurred during reconstruction.

Chapter 4

Construction of textured 3D model

The aim of this project was to construct 3D models of scenes from one image. The construction of models contains two problems: localization of model points in the 3D space and photo realistic rendering of surfaces. In this chapter we describe our approach to these problems.

There are many approaches to the construction of 3D models. Having calibrated the camera, we can easily obtain projection rays along which model points are situated. The problem of 3D reconstruction lies in finding exact positions of points on projection rays. Most of the research that has been done in this domain uses two or more images to obtain positions of model points in space. For single images Peter Sturm and Stephen Maybank in [SM99a] propose a method based on sets of planes and sets of points linked together. Antonio Criminisi and Andrew Zisserman proposed another approach in [CRZ99]. It uses measurements on planes and lines for sequential reconstruction of 3D coordinates of points.

In our work we used the idea introduced in [CSC99] for computing relative depths of parallelepiped's corners. We evaluated a method of reconstruction of scenes consisting of a set of connected parallelograms, which is described in the first section.

The last stage of the project was the extraction of photometric information. Textures obtained by simple interpolation of colors on surfaces have usually the perspective deformation. In the second section of this chapter we

describe our approach to obtaining non-deformed, photorealistic rendering of models.

Finally, we present results of some reconstructions from images of outdoor and indoor scenes.

4.1 3D points reconstruction

We assume that the camera is calibrated and so we know the matrix K of intrinsic camera parameters. This enables us to backproject image points to 3D along their projection rays. The problem of reconstruction is now to find their exact positions on projection rays.

A 3D point whose image is given by $\mathbf{q}$ has coordinates

$$\mathbf{Q} = \begin{pmatrix} \lambda \mathbf{q}' \\ 1 \end{pmatrix} \tag{4.1}$$

where $\mathbf{q}' \sim K^{-1}\mathbf{q}$ and $||\mathbf{q}'|| = 1$. $\mathbf{q}'$ is called the *viewing direction*. The unknown λ expresses the distance of $\mathbf{Q}$ form the optical center and hence defines its position on the projection ray (Fig. **??**).

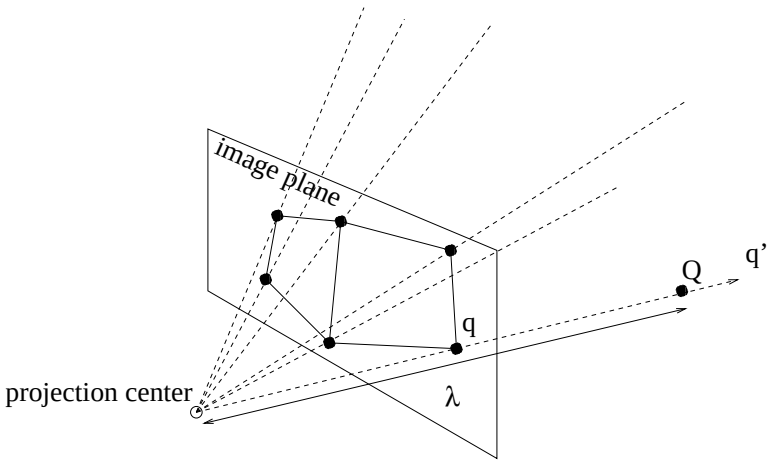


Figure 4.1: Viewing directions of image points.

For every parallelogram in space defined as shown in Figure **??** we have the following relation:

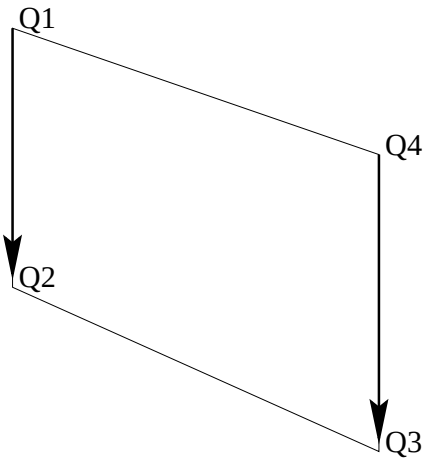


Figure 4.2: Constraint on 3D positions of parallelogram vertices.

$$\mathbf{Q}_1 - \mathbf{Q}_2 + \mathbf{Q}_3 - \mathbf{Q}_4 = \mathbf{0} \tag{4.2}$$

This implies that directions $\mathbf{q}'_i$ the parallelogram's vertices provide three equations for points depths $\lambda_1, \lambda_2, \lambda_3, \lambda_4$:

$$\lambda_1 \begin{pmatrix} q'_{11} \\ q'_{12} \\ q'_{13} \end{pmatrix} - \lambda_2 \begin{pmatrix} q'_{21} \\ q'_{22} \\ q'_{23} \end{pmatrix} + \lambda_3 \begin{pmatrix} q'_{31} \\ q'_{32} \\ q'_{33} \end{pmatrix} - \lambda_4 \begin{pmatrix} q'_{41} \\ q'_{42} \\ q'_{43} \end{pmatrix} = 0 \tag{4.3}$$

If we fix one depth, we obtain remaining three depths. Also, for a set of parallelograms connected by at least one point, it is sufficient to set arbitrarily one depth to compute the positions of all vertices of parallelogram. If we know at least one length in the scene we can fix the depths to obtain the euclidean model preserving not only ratios of lengths, but also their absolute values.

In our method we create a two directional graph, whose nodes are points of parallelograms and connections are edges of parallelograms. Next we choose the largest connex subset of the graph and we enumerate nodes. Now for each parallelogram, whose vertices were included in the graph, the equa-

tion of type (??) is added into a linear equation system:

$$\begin{pmatrix} q'_{11} & -q'_{21} & q'_{31} & -q'_{41} & 0 & 0 & 0 & 0 & ... \\ q'_{12} & -q'_{22} & q'_{32} & -q'_{42} & 0 & 0 & 0 & 0 & ... \\ q'_{13} & -q'_{23} & q'_{33} & -q'_{43} & 0 & 0 & 0 & 0 & ... \\ 0 & 0 & 0 & q'_{41} & -q'_{51} & q'_{61} & q'_{71} & 0 & ... \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \end{pmatrix} \quad (4.4)$$

After fixing one depth the resolution of the equation system above brings us information about depths of all points belonging to the set.

4.2 How to use photometric data for photo realistic rendering

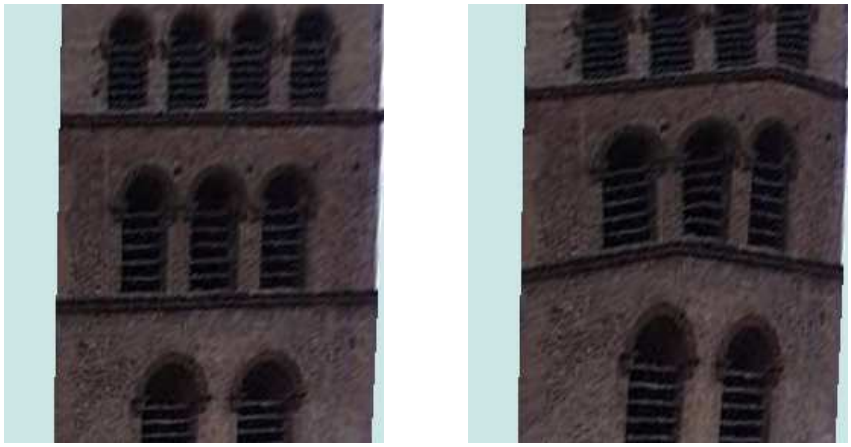
The last stage of 3D model acquisition is the extraction of photometric information. For this purpose we divided reconstructed parallelograms into triangles, adding triangles not belonging to any parallelogram, but with reconstructed vertices. Textures obtained by simple interpolation of colors in triangles in the image and triangles in the model are deformed. This is due to the fact, that the transformation from the scene object to the image is perspective, and the transformation from the triangle in the image to the triangle in the 3D model is linear. Thus without any correction we can not reconstruct the real pattern.

There are few approaches to this problem. We can for example precompute textures for each face to obtain the view as taken by a camera whose optical axis is parallel to the face's normal. In order to do this we have to compute the rotation between the optical axis (in this work we assume that it is the z-axis) and the face's normal. This allows to find the homography between the polygon on the image and polygon on new texture.

To obtain photorealistic renderings we applied a method of texture projection on the model. Texture parameters of pixels are obtained by projec-

tions of model points to the image using calibrated camera parameters.

The difference between rendering with and without the perspective correction is shown on the Figure ??.



(a) Model with texture correction.

(b) Model without texture correction.

Figure 4.3: Example of texture correction.

4.3 Experimental results

To test our methods, we constructed several models from photos of real scenes. We used both outdoor and indoor scenes. In this section we shortly describe characteristics of every scene, give examples of problems which can occur during reconstruction and present our results. Each reconstruction was made with the assumption, that the principal point lies in the center of the image.

- Our first reconstruction was the Sioux building. It is an example of a simple country house. If we do not need very detailed reconstruction we can define main object by only two primitives. It means that reconstruction is very quick. As is shown on the Figure ?? angles are nicely reconstructed. At the beginning we had strong aliasing effect,

especially on the roof. But used method of projection of the texture on the model reduce this effect.

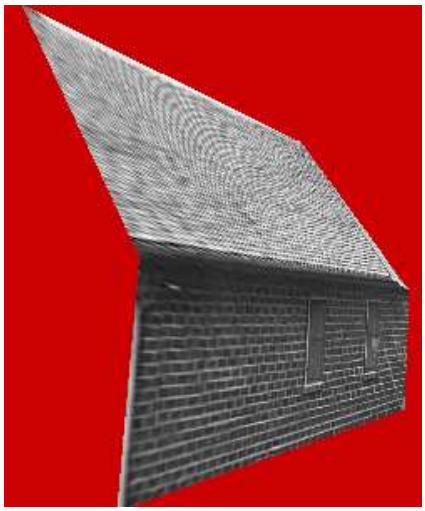
- The Notre-Dame place in Grenoble. It was defined as cuboid (the tower) and set of parallelograms. During definition of the model we had to define very carefuly the face between the tower and houses. It is very narrow, so quite sensitive to incorrect data. The second difficulty was the fact that angles are not really straight in the real scene- walls of houses are not really flat, and also fronts of houses are not really parallel to the front of the tower. But as in the previous reconstruction we see (Fig. **??**), that reconstructed angles are quite close to the reality. The photo was taken by a camera that is known to have unit aspect ratio. The value of the aspect ratio obtained by our method is 0.97, so very close to the real one.

- Torres Kio from Puerta Europa in Madrid. This is an example of a modern structure which cannot be defined by a cuboid. For calibration we used only two constraints provided by two pairs of perpendicular directions. The relative error obtained for reconstructed angles was less than 3%, and less then 2% for lengths' ratios. Some views of the scene are shown on Figure **??**.

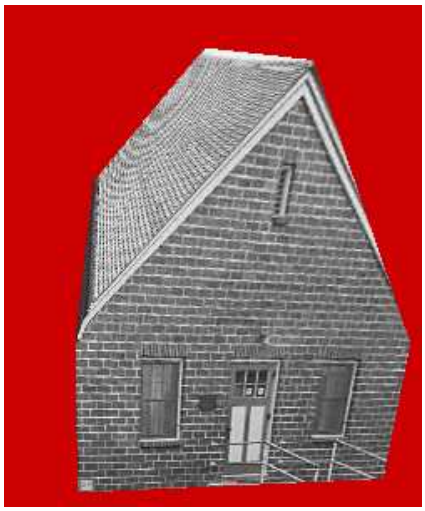
- The indoor scene of the room under the roof (Fig. **??**). To define control points of cuboids we used control marks. This photo was taken with the smallest focal of the camera. In result, we can see the effect of distortion on the image. It caused several problems with colinearity of points and coplanarity of faces in reconstructed model (Figure **??**). But general geometry of the scene is nicely reconstructed. Also textures, even on faces not very well seen by the camera, are well reconstructed.

Generally we can see on figures above, that angles are usually well reconstructed. What is also important, that we can obtain good slide of the model even from direction very different from the optical axis of a camera.

(a)



(b)



(c)

(d)

Figure 4.4: (a)-the original image of the Sioux building; (b)-(c)-(d)-photos from reconstruction

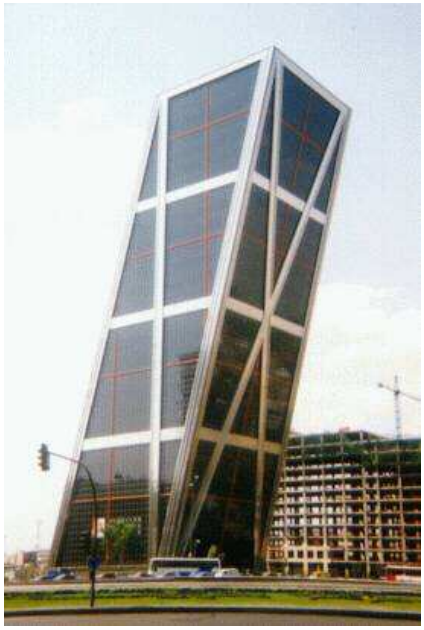
(a)

(b)

(c)

(d)

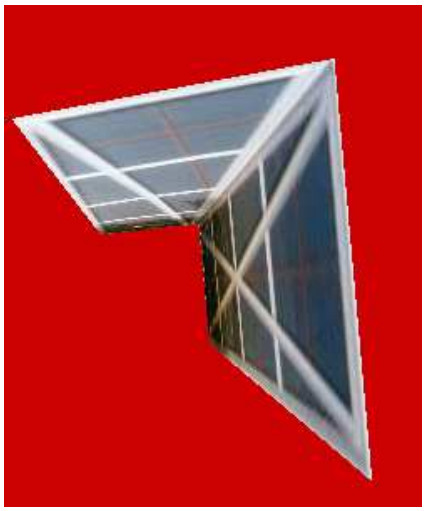
Figure 4.5: (a)-the original image of Notre Dame place; (b)-(c)-(d)- photos from reconstruction



(a)

(b)

(c)



(d)

Figure 4.6: (a)-the original image of Torres Kio; (b)-(c)-(d)- photos from reconstruction

(a)

(b)

(c)

(d)

Figure 4.7: (a)-the original image of indoor scene;(b)-(c)-(d)-photos from reconstruction

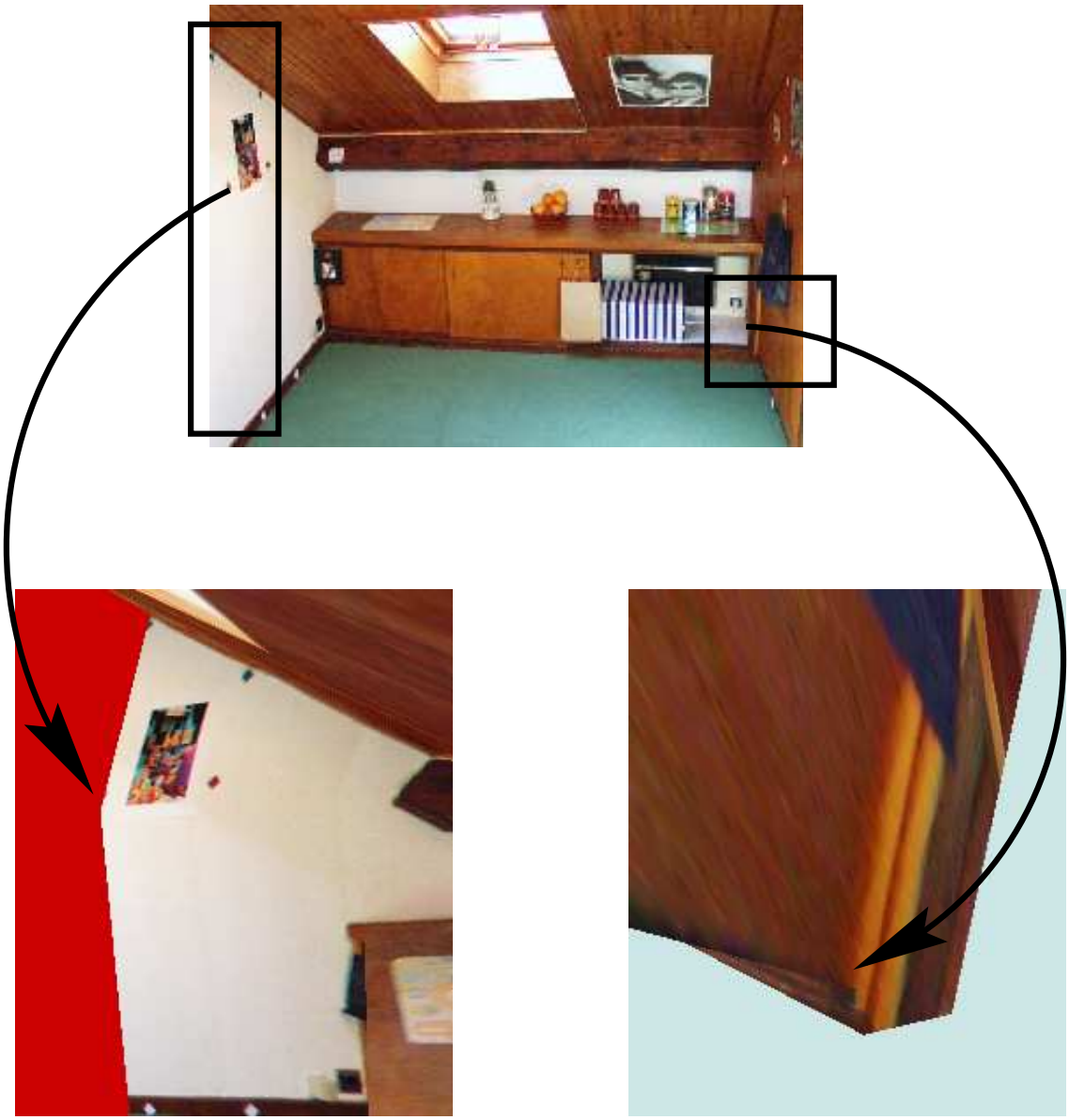


Figure 4.8: (a)-the original image of indoor scene;(b)-(c)-(d)-photos from reconstruction

Chapter 5

Conclusion

In this work we have applied computer vision theory to develop a method of 3D model acquisition from single images. We were interested mostly in outdoor and indoor urban scenes.

In case of a single image, we need additional information about the topology of the scene to be able to calibrate the camera and then construct the euclidean 3D model. In our approach, to obtain the necessary data, we employed primitives appearing in architectural environments. The most frequent and very useful primitives are parallelepipeds and parallelograms.

At the first stage we studied constraints on camera intrinsic parameters provided by prior knowledge about parameters of parallelepipeds appearing in the scene. The method of camera calibration was based on the method of calibration based on cuboids proposed by Fabien Vignon and Edmond Boyer in [Vig99]. Our method allows using not only constraints provided by perpendicular directions, but all the information about lengths and angles of parallelepipeds. One completely defined primitive allows for complete calibration of the camera. It permits also to calibrate the camera in otherwise degenerated cases. Experimental tests of calibration brought good results even in the presence of noisy data.

In the second stage we developed a method for reconstructing three-dimensional models based on sets of connected parallelograms. It occurs that such structures are sufficient to define many urban scenes. Results of reconstructions of models from photos of real scenes demonstrate good performance of our method.

Our methods have been implemented in C++. We created three independent modules: calibration, reconstruction and model visualization. The module for photorealistic rendering of the three-dimensional model was implemented using OpenGL.

There are several extensions that may be added to our method. For example other primitives can be added, like cuboids with triangle prisms on the top, spheres or cylinders. On the other hand an interactive system assisting in definition of additional primitives in partially reconstructed scenes could be developed. Also, interactive methods for combined reconstruction of two non intersecting sets of connected parallelograms could be developed. Interesting results could be obtained by joining models reconstructed from photos of different parts of a scene. Dependencies provided by known relative positions between primitives and projections of one primitive in several cameras could be studied as well.

Bibliography

[CB98] R. Cipolla and E. Boyer. 3d model acquisition from uncalibrated images. In *IAPR Workshop on Machine Vision Applications, Chiba, Japan*, pages 559–568, November 1998.

[CRZ99] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. In *Proc. 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 434–442, September 1999.

[CSC99] Y-P Hung C-S Chen, C-K Yu. New Calibration-free Approach for Augmented Reality Based on Parametrized Cuboid Structure. *International Conference on Computer Vision*, 4:30–37, 1999.

[CT90] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration. *IJCV*, 4:127–139, 1990.

[Fau92] O. Faugeras. What can be seen in three dimensions with an uncalibrated stere o rig? In G. Sandini, editor, *ECCV92*, pages 563–578. sv, may 1992.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* Artificial Intelligence. MITP, 1993.

[Har95] R. Hartley. In defence of the 8-point algorithm. In *ICCV95*, pages 1064–1070, juin 1995.

[HGC92] R.I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *CVPR92*, pages 761–764, 1992.

[PTVn92] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Fla nnery. *Numerical Recipes in C - The Art of Scientific Computing.* cup, 2nd edition, 1992.

[SK52] J.G. Semple and G.T. Kneebone. *Algebraic Projective Geometry.* osp, 1952.

[SM99a] P. Sturm and S.J. Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *BMVC - 10th British Machine Vision Conference, Nottingham, England*, pages 265–274, September 1999.

[SM99b] P. Sturm and S.J. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *CVPR - IEEE International Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado*, pages 432–437, June 1999.

[Vig99] Fabien Vignon. Modelisation de scenes urbaines a partir d'un ensemble d'images. Technical report, I.N.S.A. - E.N.S. - U.C.B.L., 1999.