CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY

Master's Thesis

Photogrammetric Method for Measuring Breathing Movements

Jiří Čihák

jiri@cihak.cz

January 2005

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/users/sara/cihak-master2005.pdf

Thesis Advisor: Dr. Ing. Radim Šára

This work has been supported by the Internal Grant Agency of
Ministry of Health of the Czech Republic under grant NK/7735-3

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Anotace

V této práci je popsána aplikace fotogrametrické metody pro měření dýchacích pohybů hrudníku a zad měřeného subjektu. Metoda je navržena pro účely klinické studie, která vyhodnocuje úspěšnost rehabilitační léčby.

Měřená místa na lidském hrudníku jsou označeny speciálními, jednoduše detekovatelnými značkami. Kalibrovaný systém se šesti kamerami snímá obrazy měřeného subjektu během dýchání. Systém je schopen značky v obrazech automaticky detekovat a následně výpočítat jejich 3D polohu fotogrametrickou metodou.

Poloha měřených značek v pracovní prostoru zorných úhlů kamer je vypočítána s prostorovou odchylkou přibližně ±0.1mm na vzdálenost jednoho metru.

Abstract

This thesis describes a system for measuring human breathing movements by a photogrammetric method. This method is aimed at quantitative evaluation of rehabilitation treatment efficiency.

The points of interest on human trunk surface are marked by special simply detectable marks. The calibrated system includs six photogrammetric cameras that grab images of measured the subject during his free breathing. The marks are automatically detected in images and their 3D positions are computed by a photogrammetric method.

The measuring system is able to compute 3D positions of special marks inside working space of the cameras with spatial accuracy of approximately ±0.1mm at 1m measuring distance.

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně s přispěním vedoucího práce a použil jsem pouze podklady (literaturu, projekty, SW atd.) v práci uvedené.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

v Praze dne ....

Jiří Čihák

Acknowledgements

I would like to thank especially to my supervisor, Dr. Radim Šára, who initiated me into the project. Without him this thesis would not exist. Through my participation in this project he expanded my skills in task solving. His willingness, support, and time devoted to me were an encouragement during my working on this project.

I would like to thank to Ing. Vadimír Smutný and Ing. Petr Pohl, who supported the development of measuring hardware very actively. I wish also to express my appreciation to everybody, who participated in the measuring experiments.

Contents

Chapter 1

Introduction

This work is a part of a three-year project *Photogrammetric Method for Measuring Breathing Movements* conducted in collaboration among the Center for Machine Perception at the Czech Technical University in Prague, Department of Rehabilitation Medicine at the Postgraduate Medical School in Prague, and Department of Physiotherapy at Charles University in Prague. I started participating in this project in 2003. This thesis describes the software system for measuring breathing movements and their 3D visualizations developed during this time.

The goal of this project is to measure movements of points of interest on human trunk surface during human breathing by a photogrammetric method. This data will be used for clinical study aimed at quantitative evaluation of rehabilitation treatment efficiency.

In order exactly defined points on human trunk to be measured by a photogrammetric method, they must be marked by some simply detectable marks. These special marks are placed on the human trunk and their images are grabbed by cameras. Assuming that we have detected positions of every mark in at least two images, its 3D position can be reconstructed by triangulation. This way we are able to obtain 3D positions of each mark at image acquisition rate during free breathing movements.

1.1 Image Acquisition

The measuring system consists of six cameras, mounted on a rigid rig. The measuring subject stands in the middle of the rig on a support of adjustable high, as shown in Figure 1.1. The subject's position is fixed by head and lumbar supports (not shown). Front cameras c_1, c_2, c_3, and c_4 observe the sternal side of trunk and the back cameras c_5 and c_6 observe the dorsal side. Because the sternal side is more curved than the dorsal one, the sternal side is measured by four cameras. The surface of dorsal side of human trunk is

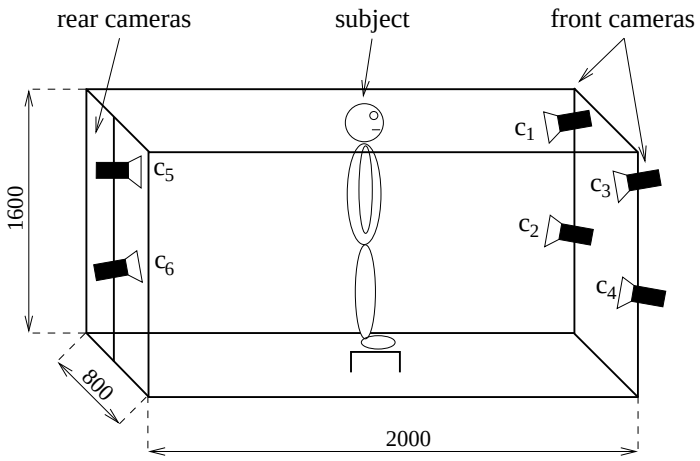


Figure 1.1: Data acquisition system

flatter, therefore two cameras suffice. Four high-frequency fluorescent tubes are mounted on the rig so that they are not in vision angles of cameras and they are lighting of human surface uniformly as well as possible (not shown).

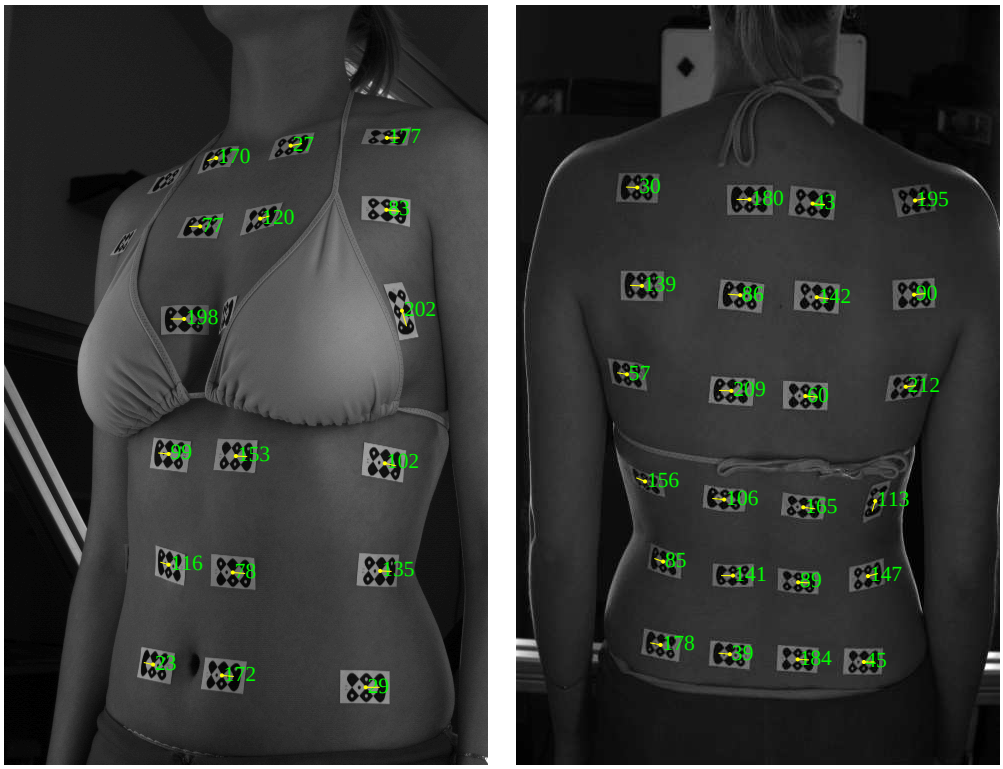
Images are obtained by six black and white 8-bit digital cameras Pulnix TM-9701 with active resolution 760×484. The camera outputs are integrated to behave as three 16-bit cameras. This is provided by special synchronization hardware. Synchronized data are then grabbed by three 16-bit digital frame grabbers DT3157 made by Data Translation and saved in the host computer as image files for subsequent processing. This hardware supports acquisition rate of 15 frames per second (one frame consisting of six images). This hardware had been developed before I started my work on the project.

1.2 Data Processing

Input to data processing software are image files obtained by image acquisition. At the beginning of each measurement session, the cameras must be calibrated. The calibration is a very important stage because the required spatial accuracy of measurement is approximately ± 0.1mm at 1m measuring distance.

The cameras are calibrated from known corresponding points in the workspace and in the images. Corresponding points are detected form images by means of a special calibration object, for which placement of calibration points is known in the world coordinate system. From corresponding points, projection matrices are computed for every camera. Once the projection matrices are established, the calibration procedure is done.

Special marks are automatically detected in each of the six grabbed images. Matching of marks is done by detection their 8-bitcode, which is implemented



(a) front Panda marks (b) back Panda marks

Figure 1.2: Detected Panda marks on human trunk

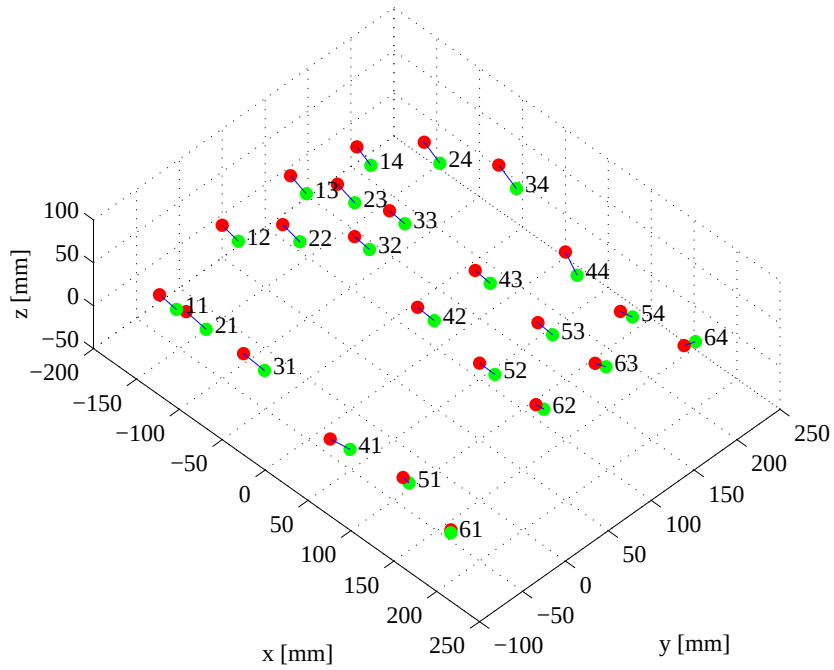
as a dot code (see sheet of Panda marks in Figure 3.3). As an example, the measured subject with detected marks is shown in Figure 1.2, where their detected bitcodes are shown in decimal numbers of green colour and their orientations in image are shown by yellow lines. Using a projection model of the mark, we get corresponding positions of each mark in several images. Assuming that we have this positions and calibrated cameras, 3D positions of marks can be reconstructed by triangulation (as described in Section 2.1.2). An example of the results of reconstruction is shown in Figure 1.3, where the position of marks at inspiration is shown by red colour and of expiration it is shown by green colour. The x axis is vertical, oriented downward. The y axis is horizontal, oriented form the right side of the subject to his/her left side. The z axis is the direction of the subject's line of sight. The decimal numbers, which describe every mark are different in the detection and in the reconstruction stages (Figure 1.2 and Figure 1.3, respectively) because they are translated by matching procedure from bitcodes to human labels for better orientation (see Section 4.1).

1.3 Related Work

A similar system for measuring breathing movements based on computer vision system ELITE is described in [2] and [8]. The ELITE is an automatic motion analyser, that measures several landmarks (up to 100) on human body by four analog cameras. Two cameras are placed in front and two behind the subject. The system is able to measure 3D positions of landmarks at a rate up to 100 frames per second with spatial accuracy of ± 0.2mm. The analog signals are sent to a parallel processor that computes 2D positions of landmarks in images at real time. The matching and 3D reconstruction is computed by a PC. The label of landmark is assigned if its position is inside a constant region in image. Before matching of landmarks, 3D positions are computed. These landmarks are passive, lightweight, and small (6mm diameter) plastic hemispheres coated by reflective layer.

1.4 Thesis Outline

Camera calibration, reconstruction by triangulation, epipolar geometry, and procedure for system calibration in a unified coordinate system are described in Chapter 2. Panda mark generation and detection is described in Chapter 3. Matching of detected Panda marks is described in Chapter 4. The results of accuracy, repeatability, and failure limit experiments of Panda mark detection is described in Section 5. Measuring protocol is outlined in Chapter 6. Conclusions and final remarks are given in Chapter 7.



(a) breathing movements of sternal side

(b) breathing movements of dorsal side

Figure 1.3: 3D reconstruction of breathing movements.

Chapter 2

Calibration

2.1 Perspective Cameras and their Systems

This section describes the projection of 3D world space to 2D image plane of a perspective camera.

2.1.1 Preliminaries

Projective camera is a non-linear device. We would like to describe it as simply as possible, using linear algebra. We therefore introduce homogeneous representation of points. Every point $\mathbf{x} = (x, y)^\top$ in R^2 can be represented in homogeneous coordinates by adding a homogeneous coordinate. Such representation will be denoted as $\tilde{\mathbf{x}} = (x, y, 1)^\top$. In general, for any non-zero constant α, the points $\tilde{\mathbf{x}}$ and $\alpha\tilde{\mathbf{x}}$ are considered the same.

Geometrically, perspective camera is described by its centre of projection $\mathbf{C}$ and image plane π. Principal axis is the line perpendicular to π passing through $\mathbf{C}$, principal point $\mathbf{p}$ is the intersection of the principal axis with π, focal length f is the distance of $\mathbf{C}$ from π (see Figure 2.1). Perspective camera maps points $\tilde{\mathbf{X}}$ from 3D scene onto 2D image plane $\tilde{\mathbf{x}}$, where $\tilde{\mathbf{X}}$ is homogeneous representation of a point in R^3. This transformation can be mathematically described by camera projection matrix $\mathbf{P}$

$$\tilde{\mathbf{x}} = \mathbf{P}\tilde{\mathbf{X}}. \tag{2.1}$$

This equation can be rewritten element-wise as

$$\begin{bmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{2.2}$$

where $u = \frac{\bar{u}}{\bar{w}}$, $v = \frac{\bar{v}}{\bar{w}}$ assuming that $\mathbf{x} = (u, v)^\top$ is a point in image plane and $\mathbf{X} = (x, y, z)^\top$ is a point in the world scene.

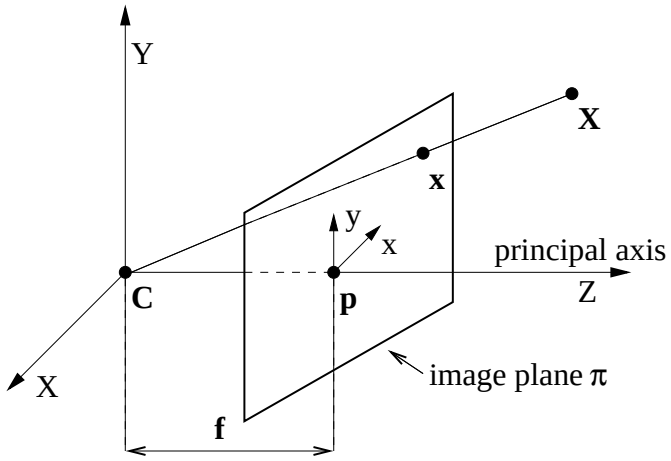


Figure 2.1: The perspective (pinhole) camera model, where **p** is the principal point, **X** is a point in 3D scene, and **x** is the projection to image plane of **X**.

Every projection matrix can be uniquely decomposed to matrix product [5, p143]

$$\mathbf{P} = \mathbf{K}\mathbf{R}\left[\mathbf{I} \;\middle|\; -\mathbf{C}\right], \tag{2.3}$$

where $\mathbf{K}$ is a 3×3 upper triangular, $\mathbf{R}$ is a 3×3 special orthogonal, and $\left[\mathbf{I} \;\middle|\; -\mathbf{C}\right]$ is a 3×4 matrix composed of a 3×3 identity matrix $\mathbf{I}$ and 3×1 vector $-\mathbf{C}$. The matrix $\mathbf{K}$ [1]

$$\mathbf{K} = \begin{bmatrix} f\,m_x & -f\,m_x \cot\Theta & x_0 \\ 0 & f\,m_y \csc\Theta & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

is called calibration matrix or matrix of intrinsic parameters of a perspective camera. In (2.4) the $\mathbf{K}$ is shown in the most general form for a finite projective camera. It defines its pixel spacing $(m_x, m_y)^T$, focal length f, skew parameter defined by the angle Θ, and the position of the principal point $(x_0, y_0)^T$. The $\mathbf{R}$ is a rotation matrix representing the orientation of the camera. The vector $\mathbf{C} = (x, y, z)^T$ is the position of the centre of projection in the world coordinate system. It is obvious that $\mathbf{P}$ has 11 degrees of freedom: 5 for $\mathbf{K}$, 3 for $\mathbf{R}$ and 3 for $\mathbf{C}$. This is the same number of freedom as a 3×4 matrix, defined up to arbitrary nonzero scale.

The matrix $\mathbf{P}$ can be decomposed to two sub matrices $\mathbf{Q}$ and $\mathbf{q}$ such that

$$\mathbf{P} = \left[\mathbf{Q}_{3\times3} \;\middle|\; \mathbf{q}_{3\times1}\right]. \tag{2.5}$$

From (2.3) it is obvious that $\mathbf{Q} = \mathbf{K}\mathbf{R}$ and $\mathbf{q} = -\mathbf{K}\mathbf{R}\mathbf{C}$, hence the centre of projection is computed as

$$\mathbf{C} = -\mathbf{Q}^{-1}\mathbf{q}. \tag{2.6}$$

More information about perspective camera can be found in [5, p136].

2.1.2 Point Reconstruction by Triangulation

Reconstruction is the task of finding a 3D point $\mathbf{X}$ in the world coordinate system given by its projections in at least two image planes. Suppose we have several images of the same point $\mathbf{X}$, then we are able to compute $\mathbf{X}$ from $\mathbf{x}_i$ and $\mathbf{P}_i$ where $\mathbf{x}_i$ are the corresponding points in image planes and $\mathbf{P}$ are the projection matrices of the cameras. We say $\mathbf{x}_i$ are in correspondence if there is a point $\mathbf{X}$ which is satisfies

$$\tilde{\mathbf{x}}_i = \mathbf{P}_i \tilde{\mathbf{X}} \tag{2.7}$$

for every $i = 1, 2, \ldots, k$ where k is the number of cameras. It means that $\mathbf{x}_i$ are in correspondence if they are induced by projections of $\mathbf{X}$.

In this section, reconstruction will be briefly explained for two calibrated cameras. The geometric interpretation is shown in Figure 2.2. The point $\mathbf{X}$ is the intersection of back-projected rays $\mathbf{r}_1$ and $\mathbf{r}_2$ induced by points $\mathbf{x}_1$ and $\mathbf{x}_2$. Provided that the points are in correspondence the rays $\mathbf{r}_1$ and $\mathbf{r}_2$ have just one intersection. In practice, it is not possible because images are sampled by pixel spacing, hence their location is not known precisely.

The equations

$$\tilde{\mathbf{x}}_1 = \mathbf{P}_1 \tilde{\mathbf{X}}_1 \quad \text{and} \quad \tilde{\mathbf{x}}_2 = \mathbf{P}_2 \tilde{\mathbf{X}}_2 \tag{2.8}$$

may be easily rewritten as a linear homogeneous system of equations. If the matrix $\mathbf{P}_j$ is given by row vectors $\mathbf{p}_{ij}^\top$ then we may rewrite (2.1) to

$$\alpha_j \begin{bmatrix} u_j \\ v_j \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{j1}^\top \\ \mathbf{p}_{j2}^\top \\ \mathbf{p}_{j3}^\top \end{bmatrix} \tilde{\mathbf{X}}, \quad j = 1, 2. \tag{2.9}$$

We can solve the last equation from (2.9) for α_j and substitute the result to the first two. We get

$$\begin{aligned} u_j \, \mathbf{p}_{j3}^\top \tilde{\mathbf{X}} - \mathbf{p}_{j1}^\top \tilde{\mathbf{X}} = 0, \\ v_j \, \mathbf{p}_{j3}^\top \tilde{\mathbf{X}} - \mathbf{p}_{j2}^\top \tilde{\mathbf{X}} = 0. \end{aligned} \tag{2.10}$$

For both corresponding points $\tilde{\mathbf{x}}_1 = (u_1, v_1, 1)^\top$, $\tilde{\mathbf{x}}_2 = (u_2, v_2, 1)$ and projection matrices given by row vectors $\mathbf{p}_{1i}^\top$, $\mathbf{p}_{2i}^\top$ this becomes a linear homogeneous system of equations

$$\begin{bmatrix} u_1 \, \mathbf{p}_{13}^\top - \mathbf{p}_{11}^\top \\ v_1 \, \mathbf{p}_{13}^\top - \mathbf{p}_{12}^\top \\ u_2 \, \mathbf{p}_{23}^\top - \mathbf{p}_{21}^\top \\ v_2 \, \mathbf{p}_{23}^\top - \mathbf{p}_{22}^\top \end{bmatrix} \tilde{\mathbf{X}} = \mathbf{0}, \tag{2.11}$$

which can be rewritten as

$$\mathbf{A}\tilde{\mathbf{X}} = \mathbf{0}, \tag{2.12}$$

where $\mathbf{A}$ is a 4×4 matrix with right null-space. Since $\tilde{\mathbf{X}}$ is a homogeneous vector, we can require the condition $\|\tilde{\mathbf{X}}\| = 1$ to find the solution. For singular

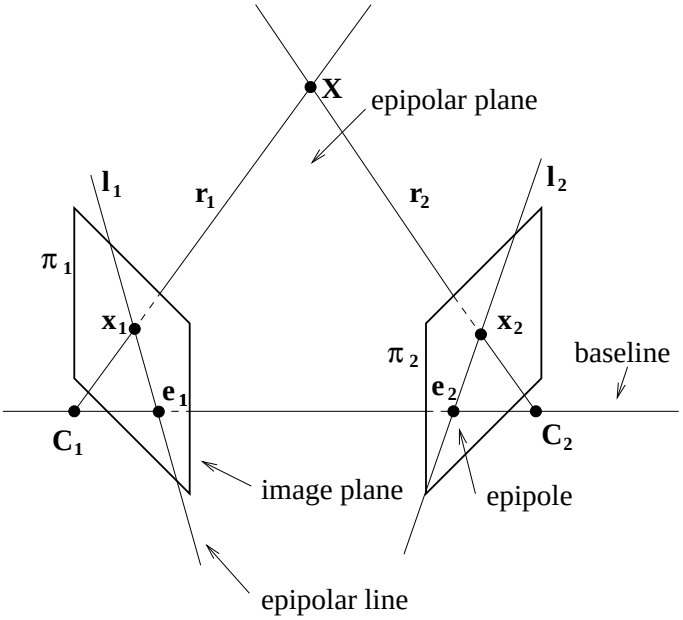


Figure 2.2: Epipolar geometry, where the baseline is the line joining the centres of projection, the epipole **e** is the intersection of the baseline and the image plane, the epipolar plane is defined by the baseline and 3D point **X**, an epipolar line **l** is the intersection of the epipolar plane and image plane.

matrix **A** with one dimensional null-space the $\tilde{\mathbf{X}}$ is unique. In the case when **A** is regular, we can use singular value decomposition SVD [3]

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top, \tag{2.13}$$

where **U** and **V** are orthogonal and $\mathbf{D} = \mathrm{diag}(\sigma_1, \ldots, \sigma_{12})$ is diagonal with non-negative elements such that $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_{12}$. The closest singular matrix in the sense of L_2 norm is then $\mathbf{A}^* = \mathbf{U}\mathbf{D}^*\mathbf{V}^\top$, where $\mathbf{D}^* = \mathrm{diag}(\sigma_1, \ldots, \sigma_{11}, 0)$. Such decomposition gives the the last column vector **V** as the solution.

2.1.3 Epipolar Geometry

Epipolar geometry is the projective relation between two calibrated cameras (see Figure 2.2). It is described by a 3×3 fundamental matrix **F** of rank 2. The **F** may be computed from intrinsic parameters of calibrated cameras or from at least 8 corresponding image points. In this section we assume that the projection matrices are known, therefore we do not need any correspondences.

If $\mathbf{x}_1$ and $\mathbf{x}_2$ are corresponding points then they are generated by same 3D point **X**. Such $\mathbf{x}_1$ and $\mathbf{x}_2$ must satisfy the epipolar constraint [5]

$$\tilde{\mathbf{x}}_2^\top \mathbf{F} \tilde{\mathbf{x}}_1 = 0, \quad \text{also} \quad \tilde{\mathbf{x}}_1 \mathbf{F}^\top \tilde{\mathbf{x}}_2^\top = 0. \tag{2.14}$$

Geometrically, the (2.14) is testing if $\mathbf{x}_2$ lies on epipolar line $\mathbf{l}_2$. The epipolar lines $\mathbf{l}_1$ and $\mathbf{l}_2$ are defined by points and $\mathbf{F}$ as follows

$$\mathbf{l}_2 = \mathbf{F}\mathbf{x}_1, \quad \text{and} \quad \mathbf{l}_1 = \mathbf{F}^\top \mathbf{x}_2. \tag{2.15}$$

The epipole $\mathbf{e}$ is defined as the projection of the centre of projection $\mathbf{C}$ of the other camera to image plane of the first camera, i.e. ,

$$\mathbf{e}_1 = \mathbf{P}_1 \begin{bmatrix} \mathbf{C}_2 \\ 1 \end{bmatrix}, \quad \mathbf{e}_2 = \mathbf{P}_2 \begin{bmatrix} \mathbf{C}_1 \\ 1 \end{bmatrix}. \tag{2.16}$$

The fundamental matrix $\mathbf{F}$ may be derived from the projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$, see [5]:

$$\mathbf{F} = -\mathbf{Q}_2^T \mathbf{Q}_1 \left[\mathbf{e}_1\right]_\times, \tag{2.17}$$

where $\left[\mathbf{e}\right]_\times$ is a 3×3 antisymmetric matrix of rank 2. It represents cross product by vector $\mathbf{e} = (e_1, e_2, e_3)^\top$ using matrix product:

$$\left[\mathbf{e}\right]_\times = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}, \quad \operatorname{rank}\left[\mathbf{e}\right]_\times = 2. \tag{2.18}$$

The epipolar constraint (2.14) and the epipolar lines (2.15) are often used as matching constraints in multiple view geometry. This will be used in Section 4.2.

2.2 The Camera Calibration Problem

Camera calibration is a process of estimating its intrinsic and extrinsic parameters. Generally, there are 6 intrinsic and 5 extrinsic parameters in projection matrix $\mathbf{P}$. In practice, we are often not estimating the parameters but directly the whole $\mathbf{P}$ is computed from (2.1) given sufficient number of corresponding points $\mathbf{x}$ in image plane and $\mathbf{X}$ in world coordinates. This will be described in the next section.

2.2.1 Camera Calibration

In Section 2.1.2 we derived (2.10) from (2.1). In this case the points $\mathbf{x}$ and $\mathbf{X}$ are known and we are estimating the elements of matrix $\mathbf{P}$ given by row vectors $\mathbf{p}_i^\top$. We may convert (2.10) to a linear homogeneous system of equations

$$\begin{bmatrix} \tilde{\mathbf{X}}^\top & \mathbf{0}^\top & -u\tilde{\mathbf{X}}^\top \\ \mathbf{0}^\top & \tilde{\mathbf{X}}^\top & -v\tilde{\mathbf{X}}^\top \end{bmatrix} \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix} = \mathbf{0}, \tag{2.19}$$

which can be rewritten as

$$\mathbf{A}\mathbf{p} = \mathbf{0}, \tag{2.20}$$

where $\mathbf{A}$ is a $2n \times 12$ matrix for the set of n points correspondences. Since $\mathbf{P}$ has 12 elements and 11 degrees of freedom, it is necessary for minimal solution to have 11 equations to solve for $\mathbf{P}$. Every point correspondence leads to two independent equations, hence we need at least $5\frac{1}{2}$ correspondences. The $\frac{1}{2}$ indicates that we use one equation only from the sixth correspondence point, u or v coordinate of point $\mathbf{x}$ on image plane. In this case, $\mathbf{A}$ has one-dimensional null-space, which gives a unique solution of (2.20) assuming $\|\mathbf{p}\| = 1$.

In practice, more than $5\frac{1}{2}$ correspondences are used for estimating $\mathbf{P}$. It may be obtained by minimizing geometric or algebraic error using SVD decomposition as described in Section 2.1.2. The estimation of $\mathbf{P}$ is possible only if points $\mathbf{X}$ are linearly independent. Thus they can not lie in the same plane in world coordinate system or some of the other known degenerate configurations [5, p516].

The assumption throughout this section has been that projective camera is a linear device in projective space. Thus world lines are projected to lines in image plane. For real lenses, this assumption is not valid. The most important deviation is radial distortion. In practice, this error is more significant for shorter focal length. We denote the image coordinates of a point $\mathbf{x}'_i$ under non-distorted projection by $\mathbf{x}_i$. The correction of radial distortion [5, p178] may then be written as

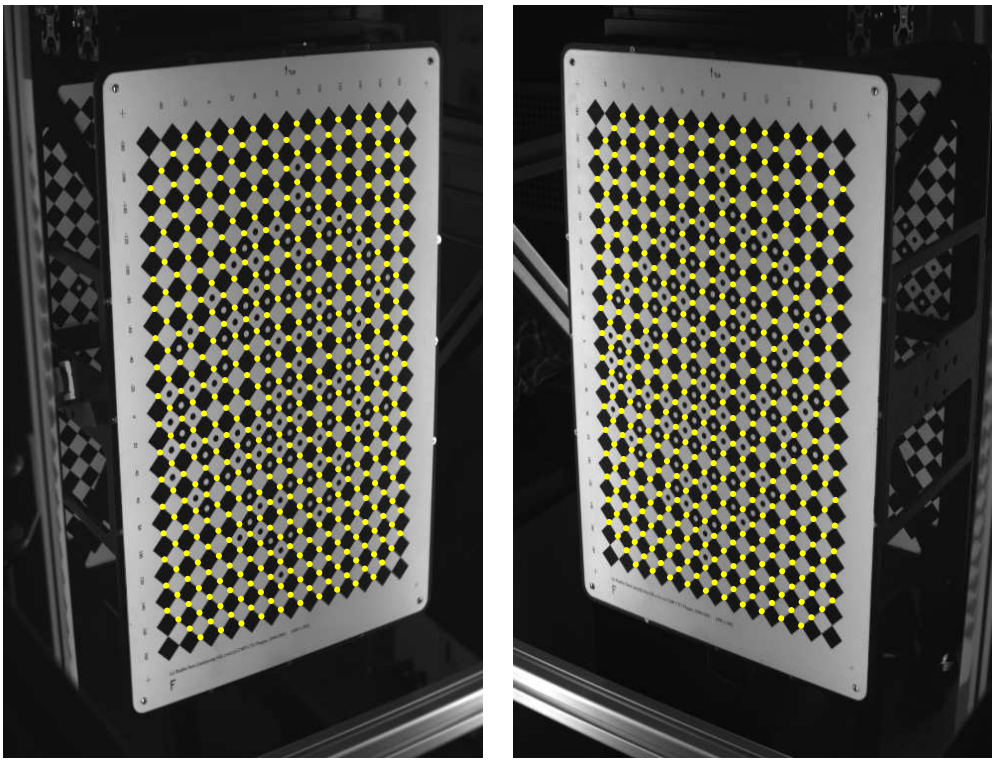
$$\mathbf{x}' = \mathbf{x}_c + L(r)\,(\mathbf{x} - \mathbf{x}_c), \tag{2.21}$$

where $\mathbf{x}$ are measured image points, $\mathbf{x}_c$ is the centre of radial distortion, $L(r)$ is a polynomial function defined as $L(r) = 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_2 r^3 + \ldots$, r is radial distance from $\mathbf{x}_c$ defined as $r = \|\mathbf{x} - \mathbf{x}_c\|$ and $\mathbf{x}'$ are corrected new image points. The estimation of coefficients $\{\mathbf{x}_c, \kappa_1, \kappa_2, \kappa_3, \ldots\}$ can be computed together with $\mathbf{P}$ during an iterative minimization of the reprojection error.

2.2.2 Photogrammetric System Calibration

System calibration is a very important stage because at the required spatial accuracy of approximately ±0.1mm at 1m measuring distance. Since the rig is influenced by many effects such as temperature dilation and mechanical distortion, calibration must be done before every measurement session.

Projection matrices $\mathbf{P}_i$ of all cameras $i = 1, \ldots, 6$ are computed by solving linear homogeneous system of equations (2.19) from corresponding points (as described in Section 2.2.1). We would like to obtain corresponding points as simply as possible, therefore a flat calibration panel with black and white chessboard pattern is used as the calibration object. The positions of calibration points in world coordinates $\mathbf{X}$ are known for a known position of the



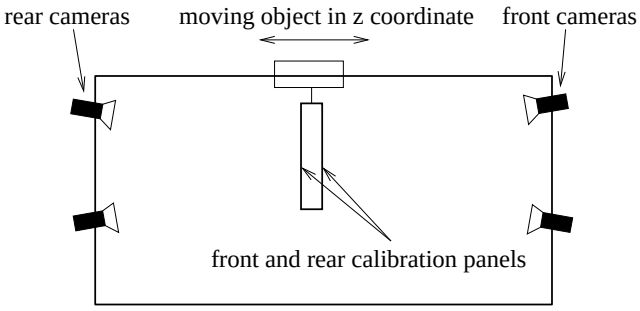
(a) left camera (b) right camera

Figure 2.3: Calibration figures with detected corresponding points $\mathbf{x}_i$

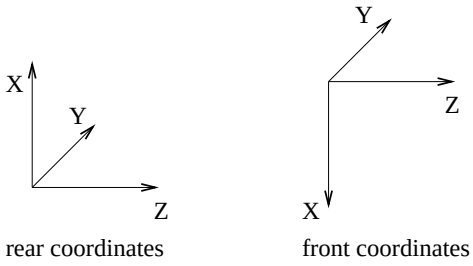
panel and the positions of their corresponding points in image planes $\mathbf{x}_i$ are easy detectable. This pattern will be called calibration figure, and is shown in Figure 2.3, where the yellow vertices are the detected calibration points $\mathbf{x}_i$. The matching of $\mathbf{x}_i$ is done by their unique code. The code for each one consists of dot code over its neighbourhood. This way we get points in image planes $\mathbf{x}_i$, which correspond to known positions of points $\mathbf{X}$ in world coordinates for each position of the plane.

Since the front and the back cameras cannot see a common plane, two calibration figures are printed on two panels mounted on a rigid steel frame. This will be called the calibration object. The figures are printed by serigraph on 1mm thick steel plate. The panels are attached to the frame. The calibration object may be moved in the z coordinate to several calibrated positions as shown in Figure 2.4(a). In this manner we get the corresponding points from more then one plane.

The process of detection calibration points $\mathbf{x}_i$ from images of calibration figure is described in detail in [9]. The detection is sufficient to obtain localization accuracy around $\frac{1}{10}$ of a pixel. Since real cameras are used, the



(a) Measuring rig with moving calibration panels (side view)



(b) Different coordinate systems

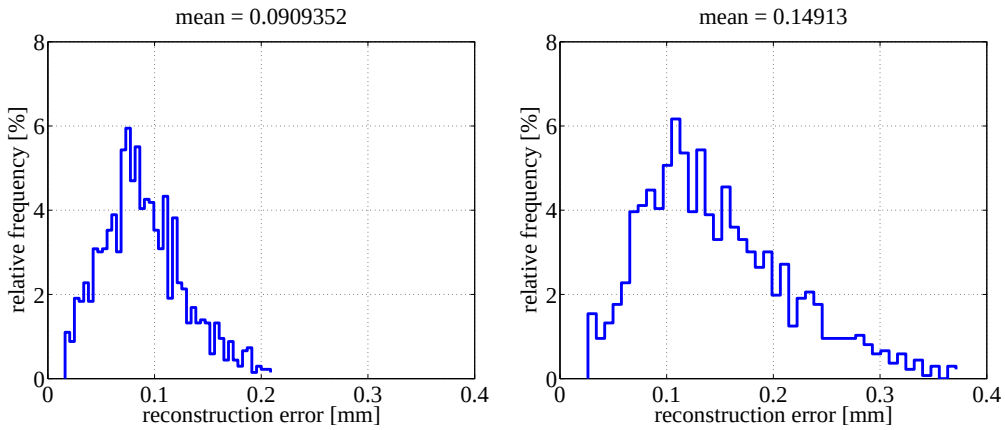
Figure 2.4: Calibration of cameras by a pair of moving panels

computation of projection matrices $\mathbf{P}_i$ is followed by bundle adjustment calibration [10] that includes a radial distortion model. Bundle adjustment [5], [12] is a non-linear constrained optimization problem that estimates parameters jointly with a correction of the calibration points. It is a non-trivial task and will not be described here.

The calibration accuracy is described by histograms of residual spatial geometric errors between points $\mathbf{X}$ and reconstructed points $\mathbf{X}'$ using computed projection matrices $\mathbf{P}_i$. Their histograms are shown in Figure 2.5 for the front and the back cameras, respectively. Mean error is smaller in the front coordinate system than in the back because the front space is measured by four cameras and the back space is measured by two cameras only.

2.2.3 Coordinate System Unification

As discussed above, camera calibration, is done in two different coordinate systems because the calibration panel has unknown width and the calibration figures are not mirrored versions of each other. We would like to establish transformation matrix $\mathbf{T}$ which maps points from the back camera coordinate system $\mathbf{O}_2$ to the front camera coordinate system $\mathbf{O}_1$. We assume we can measure a set of points $\mathbf{X}$ that are visible in front and back cameras simultaneously.



(a) Front coordinate system (b) Back coordinate system

Figure 2.5: Histograms of the residual calibration error in millimetres

By reconstruction in the front and the back coordinate systems we obtain two sets of points $\mathbf{X}_1$ and $\mathbf{X}_2$ that are related by an unknown rigid motion. The sets of reconstructed points $\mathbf{X}_1$ and $\mathbf{X}_2$ are represented by $3 \times n$ matrices, where n is the number of points. The transformation matrix $\mathbf{T}$ consists of sub matrices $\mathbf{R}$, $\mathbf{t}$ as follows

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2.22}$$

where $\mathbf{R}$ is a 3×3 special orthogonal matrix representing rotation, $\mathbf{t}$ is a translation 3-vector, and $\mathbf{T}$ is a 4×4 rigid motion matrix of 3D points in homogeneous coordinates. The projection matrices $\mathbf{P}_i$ of back cameras may be updated by $\mathbf{T}$ such that

$$\mathbf{P}'_i = \mathbf{P}_i \mathbf{T}, \tag{2.23}$$

where $\mathbf{P}'_i$ are the projection matrices of back cameras transformed to the front coordinate system.

The standard 3D-3D absolute orientation problem [7] is to infer $\mathbf{R}$ and $\mathbf{t}$ from a set of points $\mathbf{X}_1 = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ and $\mathbf{X}_2 = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ which satisfy equations

$$\mathbf{q}_i = \mathbf{R}\mathbf{p}_i + \mathbf{t}, \quad i = 1, \ldots, n. \tag{2.24}$$

Parameters of $\mathbf{R}$ and $\mathbf{t}$ can be estimated by the least-square method. We follow the derivation given in [7] and update their method for use with SVD. Translation $\mathbf{t}$ drops out by subtracting the mean vectors

$$\mathbf{p}'_i = \mathbf{p}_i - \overline{\mathbf{p}}, \quad \mathbf{q}'_i = \mathbf{q}_i - \overline{\mathbf{q}}, \tag{2.25}$$

where $\overline{\mathbf{p}}$ and $\overline{\mathbf{q}}$ are defined as $\overline{\mathbf{p}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i$ and $\overline{\mathbf{q}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{q}_i$. We will find $\mathbf{R}$ by

minimizing the least-square error

$$\mathbf{R} = \arg\min_{\mathbf{Q}} \sum_{i=1}^{n} \left(\mathbf{p}'_i - \mathbf{Q}\,\mathbf{q}'_i\right)^{\top} \left(\mathbf{p}'_i - \mathbf{Q}\,\mathbf{q}'_i\right), \tag{2.26}$$

where $\mathbf{Q}$ is a special orthogonal 3×3 matrix. The problem can be rewritten as

$$\mathbf{R} = \arg\min_{\mathbf{Q}} \sum_{i=1}^{n} \left(\|\mathbf{p}'_i\|^2 - 2\,\mathbf{p}'^{\top}_i \mathbf{Q}\,\mathbf{q}'_i + \|\mathbf{q}'_i\|^2\right). \tag{2.27}$$

It is obvious that least-square error is minimal if the term $2\mathbf{p}'^{\top}_i \mathbf{Q}\mathbf{q}'_i$ is maximal in the above expression, hence the $\mathbf{R}$ can be computed as

$$\mathbf{R} = \arg\max_{\mathbf{Q}} \sum_{i=1}^{n} \mathbf{p}'^{\top}_i \mathbf{Q}\,\mathbf{q}'_i. \tag{2.28}$$

Provided that [7]

$$\mathbf{p}'^{\top}_i \mathbf{R}\,\mathbf{q}'_i = \operatorname{tr}(\mathbf{R}^{\top}\mathbf{p}'_i\mathbf{q}'^{\top}_i), \tag{2.29}$$

the equation (2.28) can be rewritten to

$$\mathbf{R} = \arg\max_{\mathbf{Q}} \sum_{i=1}^{n} \operatorname{tr}\left(\mathbf{Q}^{\top}\mathbf{p}'_i\mathbf{q}'^{\top}_i\right) = \arg\max_{\mathbf{Q}} \operatorname{tr}\left(\mathbf{Q}^{\top}\mathbf{M}\right), \tag{2.30}$$

where $\mathbf{M}$ is defined as

$$\mathbf{M} = \sum_{i=1}^{n} \mathbf{p}'_i\mathbf{q}'^{\top}_i \tag{2.31}$$

Note that $\mathbf{M}$ is square positive semidefinite matrix. Therefore, if $\mathbf{M}$ is regular it exists a unique polar decomposition [4] such that

$$\mathbf{M} = \mathbf{U}\mathbf{S}, \tag{2.32}$$

where $\mathbf{U}$ is orthogonal matrix and $\mathbf{S}$ is a symmetric matrix. The $\mathbf{S}$ is uniquely given as $\mathbf{S} = (\mathbf{M}^{\top}\mathbf{M})^{\frac{1}{2}}$. Matrix $\mathbf{S}$ can be decomposed by SVD to [5]

$$\mathbf{S} = \sum_{j=1}^{3} \left(\sigma_j^2 \mathbf{u}_j \mathbf{u}_j^{\top}\right), \tag{2.33}$$

where $\|\mathbf{u}_j\| = 1$ and $\mathbf{u}_j^{\top}\mathbf{u}_k = 0$ for $j \neq k$. Formula (2.32) can be rewritten by using (2.33) and substituted to (2.30). We get

$$\mathbf{R} = \arg\max_{\mathbf{Q}} \operatorname{tr}\left(\mathbf{Q}^{\top}\mathbf{U}\sum_{j=1}^{3} \sigma_j^2 \mathbf{u}_j \mathbf{u}_j^{\top}\right) = \arg\max_{\mathbf{Q}} \operatorname{tr}\sum_{j=1}^{3} \sigma_j^2 \mathbf{Q}^{\top}\mathbf{U}\mathbf{u}_j\mathbf{u}_j^{\top}. \tag{2.34}$$

If $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{B}$ is an $m \times n$ matrix, then the following identity holds [7]

$$\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA}). \tag{2.35}$$

Using this, equation (2.34) can be transformed to

$$\mathbf{R} = \arg\max_{\mathbf{Q}} \mathrm{tr} \sum_{j=1}^{3} \sigma_j^2 \mathbf{u}_j^\top \mathbf{Q}^\top \mathbf{U} \mathbf{u}_j = \arg\max_{\mathbf{Q}} \sum_{j=1}^{3} \sigma_j^2 \mathbf{u}_j^\top \mathbf{Q}^\top \mathbf{U} \mathbf{u}_j, \tag{2.36}$$

where trace of a scalar is the scalar value itself. Since $\mathbf{Q}^\top \mathbf{U}$ is orthogonal, the sum is maximal if $\mathbf{Q} = \mathbf{U}$, hence $\mathbf{Q}^\top \mathbf{U} = \mathbf{I}$. We get

$$\mathbf{R} = \mathbf{U} = \mathbf{M} \left(\mathbf{M}^\top \mathbf{M} \right)^{-\frac{1}{2}}. \tag{2.37}$$

In the general case, the $\mathbf{M}$ may be singular. If it has rank 2 a unique solution can still be found as follows. The rotation matrix $\mathbf{R}$ is expressed by using SVD from (2.37), where $\mathbf{M} = \mathbf{WDV}^\top$ such that

$$\mathbf{R} = \mathbf{WDV}^\top \left(\mathbf{VDW}^\top \mathbf{WDV}^\top \right)^{-\frac{1}{2}} = \mathbf{WV}^\top. \tag{2.38}$$

This is unique up to the sign of the last column of $\mathbf{V}$ (or last row of $\mathbf{U}$), which corresponds to the zero singular value (changing the sign of this column vector does not change the $\mathbf{M}$, where $\mathbf{M}$ is singular of rank 2). This will be amended as follows.

The matrix $\mathbf{R}$ is guaranteed to be orthogonal, but it is not guaranteed to have determinant $+1$. Determinant -1 means that $\mathbf{R}$ incorporates a reflection. If the matrix $\mathbf{V}^\top$ is given by its row vectors $\mathbf{v}_i^\top$

$$\mathbf{V}^\top = \begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \mathbf{v}_3^\top \end{bmatrix}, \tag{2.39}$$

a solution for $\mathbf{R}$ is immediately obtained as

$$\mathbf{R} = \mathbf{W} \begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \mathbf{v}_3^\top \end{bmatrix} \quad \text{or} \quad \mathbf{R} = \mathbf{W} \begin{bmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ -\mathbf{v}_3^\top \end{bmatrix}. \tag{2.40}$$

We choose the one of equations (2.40) that gives $\det \mathbf{R} = +1$.

Once the rotation matrix $\mathbf{R}$ is known, the translation $\mathbf{t}$ can be computed from

$$\mathbf{t} = \overline{\mathbf{q}} - \mathbf{R}\overline{\mathbf{p}}. \tag{2.41}$$

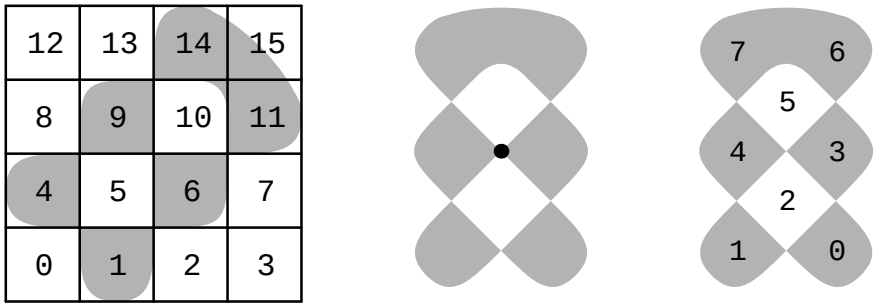
The transformation matrix $\mathbf{T}$ in (2.22) is fully defined now.

Chapter 3

Panda Marks

In this project, movements of exactly defined points on human thorax are measured, hence special marks are needed here [13]. They are called Panda marks and have been developed by Vít Zýka [13].

3.1 Description



(a) 4×4 defining grid (b) the reference point (c) bit location

Figure 3.1: Panda marks (taken from [13])

The Panda marks are designed to be detectable automatically in the image very accurately under arbitrary orientation and perspective distortion. Their shape is inscribed in a 4×4 grid, see Figure 3.1(a). The reference point is located in the middle and defines the position of the mark, see Figure 3.1(b). From Figure 3.1 it is obvious that the mark is not centrally symmetric, therefore its orientation in the image can be determined. Provided that the shape and orientation are known, the positions of bits are known as well. The position of bits are shown in Figure 3.1(c). Every Panda marks can be uniquely

identified by its 8-bit code. Thus 256 different marks can be distinguished. Its code will be used in Panda mark matching between images, which is a necessary step to compute 3D location of Panda marks.

3.2 Sheet of Panda Marks

The marks are drawn by the MetaPost language [6]. The source filename of Panda mark is `panda.mp`. The command

```
mpost panda.mp
```

generates a set of Encapsulated PostScript files. They contain all individual marks, illustrative figures, and the complete mark set. Their general filename is `panda.xxx`, where `xxx` is a number from Table 3.1.

Filename extension	**Description**
0, 1, ..., 255	individual marks
300	figure of grid numbering, see Figure 3.1(a)
301	figure of reference point, see Figure 3.1(b)
302	figure of bit coding, see Figure 3.1(c)
500	example of marks for printing on sheet

Table 3.1: Files generated by panda.mp (taken from [13])

Since the marks are to be sticked on human thorax, they must be printed on a sheet of stickers. Arrangement of the Panda mark on the sheet can be easily done by the MetaPost parametrization. The parameters in file `panda_sheet.mp` are preset for the current sheet of stickers and can be modified for a different one. Table 3.2 describes the available parameters. The Encapsulated PostScript file of the sheet of Pandas `panda_sheet.0` is generated by the command:

```
mpost panda_sheet.mp
```

Otherwise, the PostScript file can be generated as follows:

```
mpost panda_sheet.mp
latex panda_sheet.tex
dvips -T Xcm,Ycm panda_sheet.dvi
```

In the last line the media size are set, where X is paper width and Y paper height.

The generated sheet of Panda marks is shown in Figure 3.3.

Parameter	Description
`draft`	draw label borders (borders are drawn if it set to true value)
`rot`	mark rotation in degrees (horizontal orientation if set $-45°$, vertical orientation if set $45°$)
`Xnum`	number of sheet columns
`Ynum`	number of sheet rows
`firstOccup`	the first mark printed on the sheet from bottom-left corner (for partly filled sheet)
`numOccup`	maximal count of marks printed on sheet (ignored if it set to zero)
`marks`	bitcodes for each mark, string of hexadecimal numbers separated by a space characters (e.g. `"00 0A 0B FF"`)
`labels`	human labels of each mark, string of hexadecimal numbers separated by a space characters (e.g. `"00 0A 0B FF"`)
`X`	horizontal width of the sticker
`Y`	vertical width of the sticker
`Xoffs`	horizontal offset from the left edge of the sticker
`Yoffs`	vertical offset from the bottom edge of the sticker
`Xsep`	horizontal separation of labels
`Ysep`	vertical separation of labels
`e`	defining grid size (scaling only marks without their labels)
`scale`	scaling factor of the sheet, its value is automatic computed from previous parameters such that $1.02\min(Y/(2e\sqrt{2}), X/(3e\sqrt{2}))$

Table 3.2: Available parameters in panda_sheet.mp (taken from [13])

3.3 Panda mark Detection in Images

The task of detection of Panda marks in images is to establish their image positions and their bitcodes. The process of detection is implemented in Matlab 6.5 and is provided by function `pandamarks` [13]. It is divided to several stages:

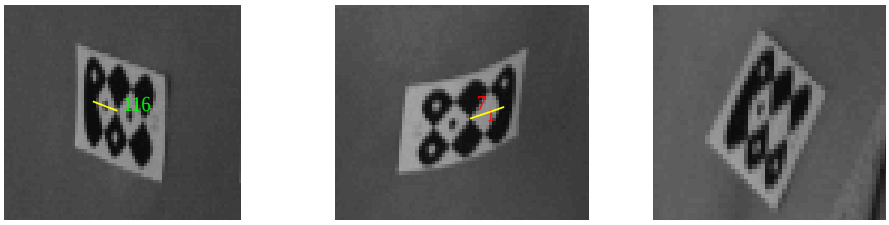
(i) Saddle point detection.

(ii) Selection of saddles whose neighbour-saddles are a in specific configuration.

(iii) Fitting a model that includes Panda mark curvature.

(iv) Computation of code bit locations.

(v) Resolution of multiple, mutually exclusive detections.



(a) successfully detected (b) detected with one indefinite bit (c) undetected

Figure 3.2: Results of Panda marks detection

(vi) Decoding of the bit code.

The output of detection are the positions of the reference points, the bit-codes as decimal numbers and the information which bit is definite or indefinite for every Panda mark. If the bit is successfully detected it is marked as definite otherwise it remains indefinite. A few real instances of Panda mark detection are shown in Figure 1.2. Three individual examples of Panda mark detection are shown in Figure 3.2, where Figure 3.2(a) illustrates a successfully detected Panda mark with all definite bits, Figure 3.2(b) illustrates a detected Panda mark with one indefinite bit and Figure 3.2(c) illustrates a undetected Panda mark.

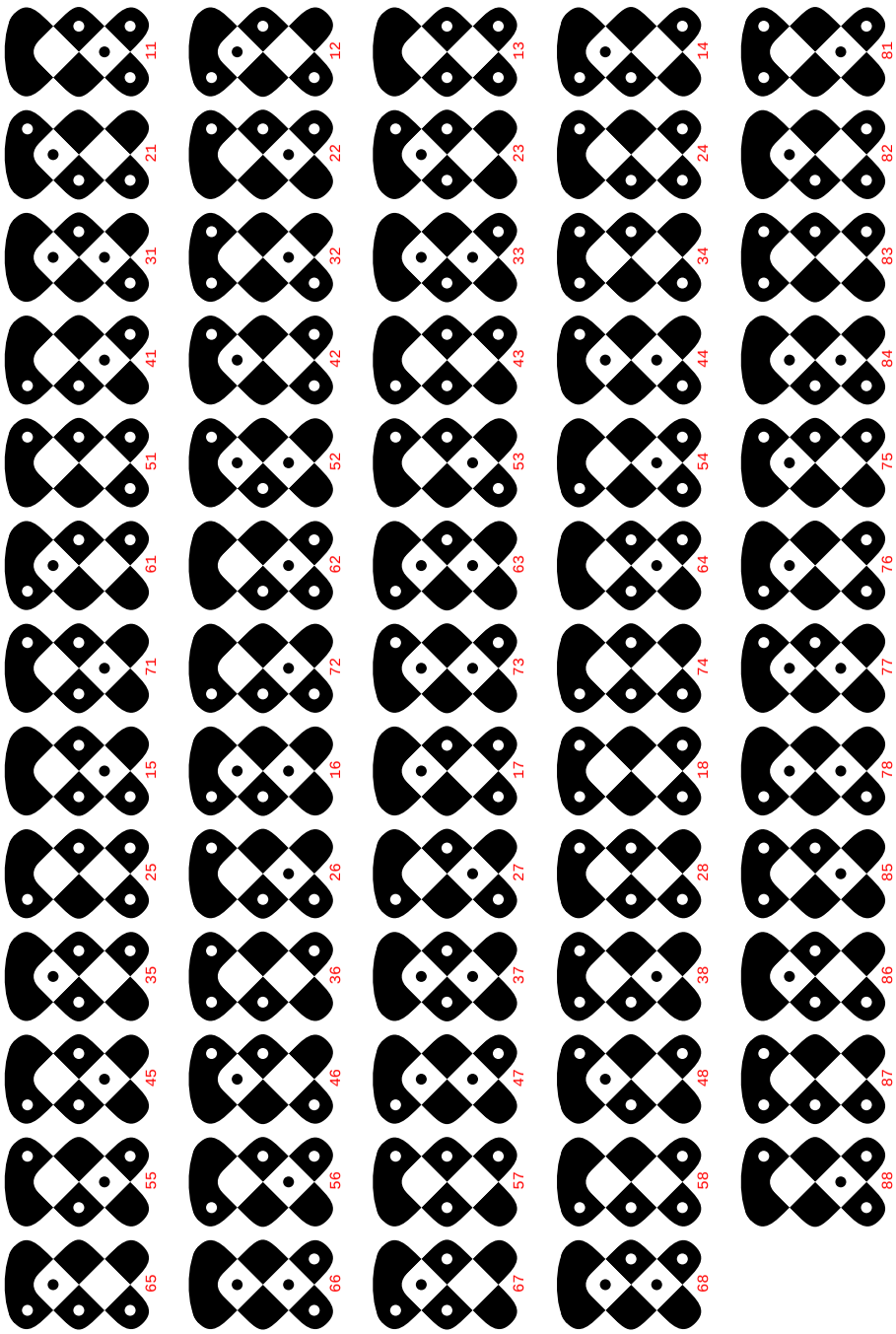


Figure 3.3: Sheet of Panda marks with their labels generates in MetaPost.

Chapter 4

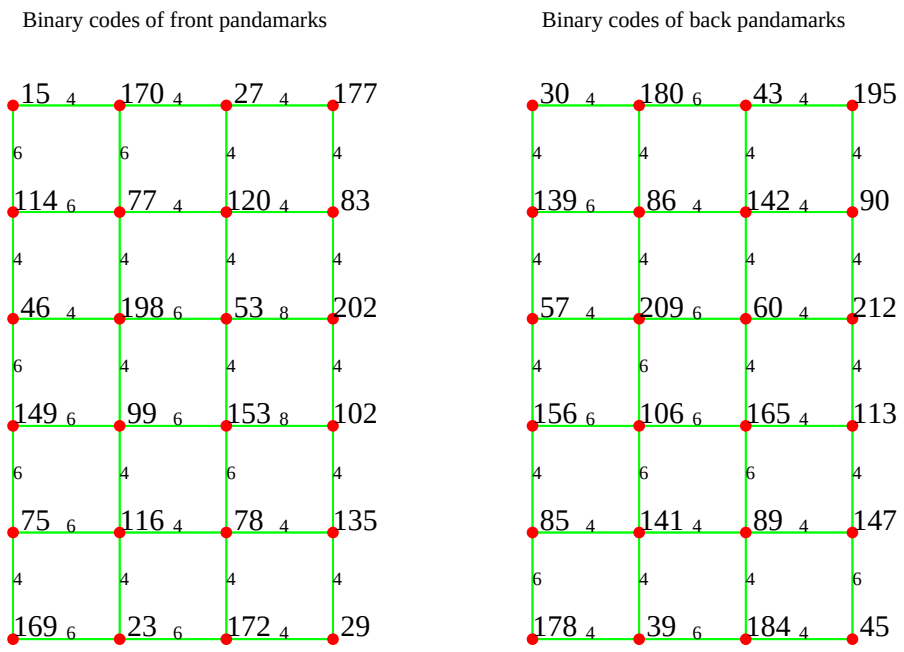
Matching

Matching is the task of finding corresponding marks in input images. Once the matching is known the 3D position can be computed by triangulation as described in Section 2.1.2. The bitcodes of marks are used for matching. If the detection is correct for each Panda mark, the solution of matching is exact and the bitcodes assignment is just processed. In practice, it is not possible because the bitcodes of Panda marks are not detected correctly in each case. The reason for this is that the marks are not flat when are sticked on the body, which may cause bitcode unreadable. Due to human thorax geometry, we need to detect marks under viewing angles as large as 70°. The correction of their bitcodes is described in this chapter.

4.1 Bitcode

Every Panda mark is uniquely identified by two different numbers: a 8-bitcode and a logical label. The 8-bitcode is written on the mark as a dot code and is detected by Panda mark detection which is described in Section 3.3. The logical label is composed of two numbers which code the position on human thorax. The transformation of bitcode to logical labels is facilitated by a transformation table. The sheet of Panda marks with their bitcodes and labels is shown in Figure 3.3.

For measuring, we use 48 marks only from all 256 available, where 24 are placed on sternal and dorsal thorax in six rows and four columns. Panda marks can be indexed by code 4 of 8, where the number of true and false values of bits are the same for each mark. The $\binom{8}{4} = 70$ different codes can be generated. The code 4 of 8 guarantees that Hamming distance is at least two between every pair of marks. This is one component of robustness in matching. The second component is based on relative placement of the marks. The mark bitcodes are assigned so that the Hamming distance of neighbours is as large as possible. The template for placement of Panda marks on human thorax is



(a) sternal side (b) dorsal side

Figure 4.1: The template of placement of code 4 of 8 on human chess

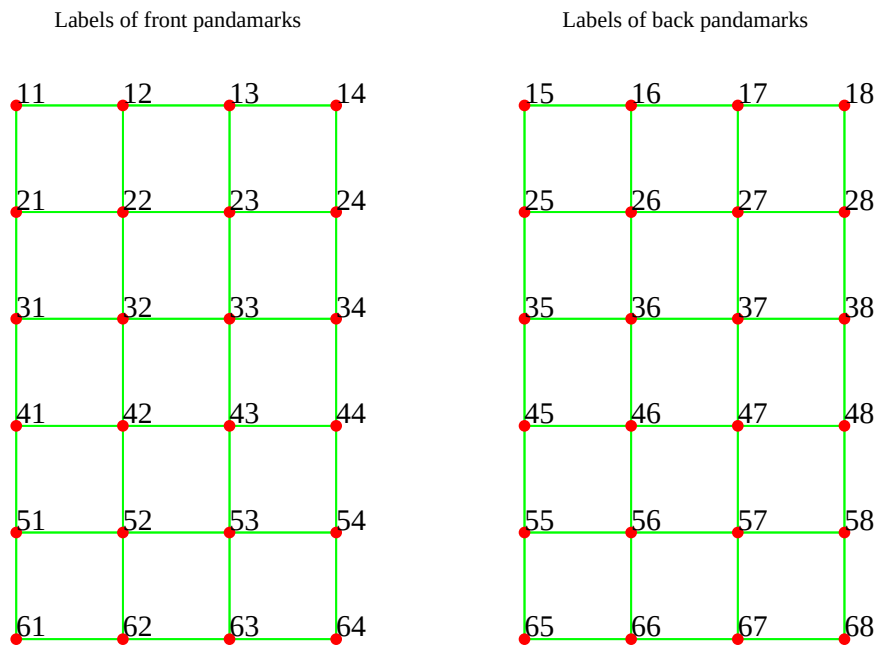
known and fixed as shown Figure 4.1. The numbers in the vertices are the Panda mark bitcodes represented as decimal numbers. The numbers in edges are Hamming distances between neighbouring bitcodes. The bitcodes of Panda marks is translated by graph matching to logical labels, as shown Figure 4.2.

4.2 Graph Matching

We wish to use as much available information for matching as possible. The bitcodes of Panda marks, their relative positions among neighbours, and epipolar geometry are employed here. This information decreases probability of mismatch and increases matching efficiency. The algorithm of graph matching consists of several parts shown in a flow diagram in Figure 4.3.

The input parameters of graph matching are the detected positions of Panda marks, their 8-bitcodes, information which bits are definite (reliably decoded) or indefinite (unreliably decoded), and a template for placement of Panda marks on human thorax, see Figure 4.1. It is assumed the definite bits are correctly decoded.

The graph matching is an iterative algorithm based on subgraph isomor-



(a) Sternal thorax (b) Dorsal thorax

Figure 4.2: The placement of Panda marks with labels on human chess

phism. It is divided to several stages. The notation and initialization of parameters are described in Section 4.2.1. The Sections 4.2.2 and 4.2.3 describe the graph matching algorithm itself.

4.2.1 Initialization

We would like to derive the information about relative neighbourhood of each Panda mark $v_i \in \mathbf{V}^c$ from their detected positions, where $\mathbf{V}^c$ is a set of detected Panda marks in image from camera c and $|\mathbf{V}^c|$ is their count. The Delaunay triangulation can be employed here. It connects points to triangles, provided that their circumscribed circle does not contain any other point. The Delaunay triangulation graph will be denoted as $\mathcal{D}^c = (\mathbf{V}^c, \mathbf{E}_D^c)$, where $v_i \in \mathbf{V}^c$ are vertices of detected Panda marks and $\mathbf{E}_D^c$ is a set of edges in graph $\mathcal{D}^c$.

We will used another graph, derived from the template (see Figure 4.1) which will be denoted as $\mathcal{T} = (\mathbf{W}, \mathbf{E}_T)$, where $w_i \in \mathbf{W}$ are vertices in the template and $\mathbf{E}_T$ is a set of edges in template $\mathcal{T}$.

The matching $\mathbf{M}$ is a collection of mappings between sets $\mathbf{V}^c$ and $\mathbf{W}$ for every c such that $\mathbf{M} = \{\mathbf{M}^c\}_{c=1}^k$ and $\mathbf{M}^c \colon \mathbf{V}^c \to \mathbf{W}$, where $v_i \in \mathbf{V}^c$ are the detected vertices and $w_i \in \mathbf{W}^c$ are the template vertices. The algorithm is

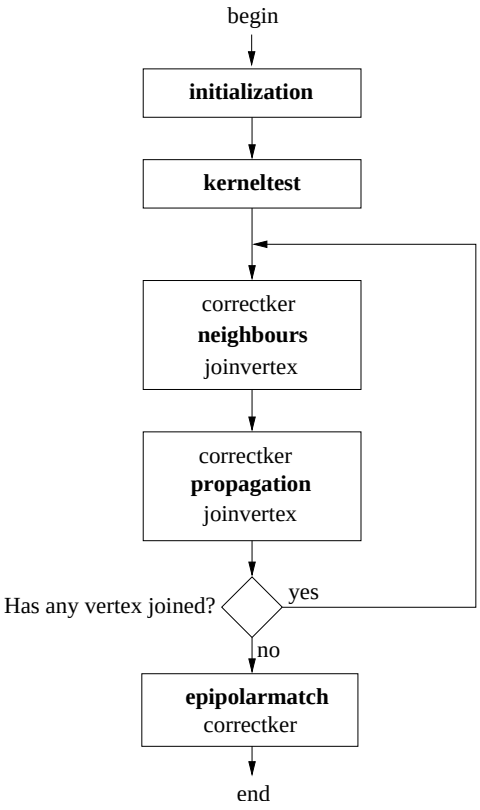


Figure 4.3: Flow diagram of graph matching.

aimed to search for maximum $\mathbf{M}$ (i.e. as large as possible).

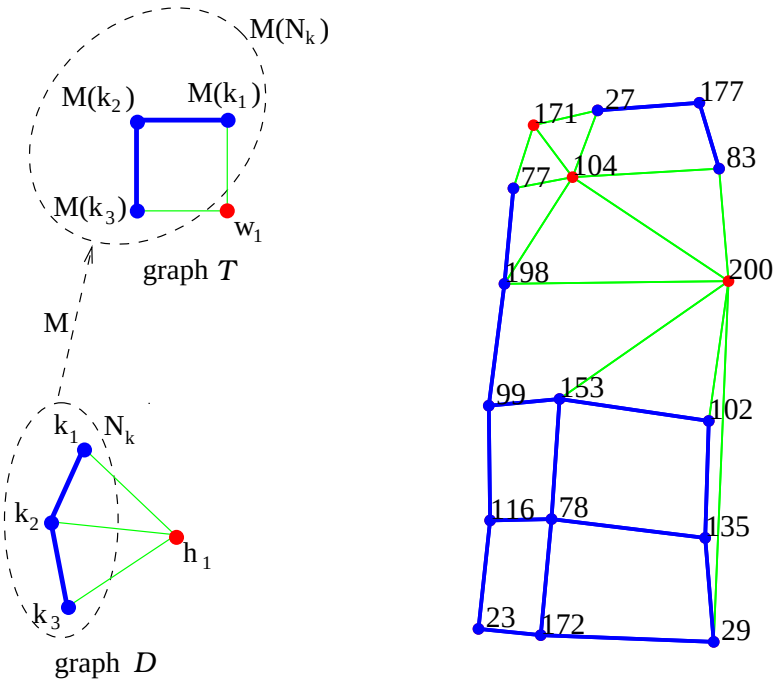
We say a vertex $v_i \in \mathbf{V}^c$ is a member of kernel $\mathbf{K}^c \subseteq \mathbf{V}^c$ (i.e. $v_i \in \mathbf{K}^c$) if:

(i) Every $v_i \in \mathbf{K}^c$ has all its bits $b(v_i)$ definite.

(ii) For every pair of vertices $v_i, v_j \in \mathbf{K}^c$ it holds $b(v_i) \neq b(v_j)$, where $b(v)$ is 8-bitcode of vertex v.

(iii) For every $v' \in \mathbf{K}^d$ in camera $d \neq c$ such that $b(v') = b(v)$, the v' must satisfy epipolar constraint with $v \in \mathbf{K}^c$. The vertices v and v' satisfy epipolar constraint if the sum of distances in pixels between their position and epipolar lines is smaller than a predefined threshold.

Once we have the successfully detected kernel vertices $\mathbf{K}^c$ for every camera c, we have a set which is consistent with the template. Every kernel vertex has a unique image in $\mathbf{W}$ via its bitcode. Hence, the mapping of kernel vertices to $\mathbf{W}$ belongs to matching $\mathbf{M}$. This is a base of matching $\mathbf{M}$. We wish to extend the matching for vertices with indefinite bits as long as they satisfy additional criteria. The next Sections 4.2.2 and 4.2.3 describe the matching algorithm



(a) Mapping M

(b) Joining of vertices 170, 120, 202

Figure 4.4: Adding vertices $\mathbf{H}$ to $\mathbf{K}$ by kernel neighbours

which proceeds by extending the set of matched vertices. The candidate for extension will be taken from a set we call a hull. We say the vertex $v \in \mathbf{V}^c$ is a member of the hull $v \in \mathbf{H}^c$ if v has any indefinite bit. As the first step, the vertices $v \in \mathbf{R}^c$, where $\mathbf{R}^c = \mathbf{V}^c \setminus \mathbf{K}^c \setminus \mathbf{H}^c$ are removed from solution. Subsequent steps are described in the following section.

4.2.2 Kernel Extension

The following procedure is done in each of camera independently, therefore we omit the superscripts describing the the cameras to simplify notation. From initialization, we get sets of vertices $\mathbf{V}$, $\mathbf{K}$, $\mathbf{H}$ and graphs $\mathcal{D}$, $\mathcal{T}$, they will be used in following algorithm.

An essential part of the algorithm is *edge update*. Edges of the subgraph of $\mathcal{D}$ induced by $\mathbf{K}$ are updated by edges of the subgraph of $\mathcal{T}$ induced by $\mathbf{M}(\mathbf{K})$ such that the new graph $\mathcal{D}'$ is obtained from $\mathcal{D}$ by the rule $\mathcal{D}' := (\mathbf{V}, \mathbf{E}_D \setminus \mathbf{E}_{DK} \cap \mathbf{E}_{TK})$, where $\mathbf{E}_{DK}$ are edges of induced graph $\mathcal{D}$ by $\mathbf{K}$ and $\mathbf{E}_{TK}$ are edges of induced graph $\mathcal{T}$ by $\mathbf{M}(\mathbf{K})$. Vertices $\mathbf{M}(\mathbf{K})$ are images of vertices $\mathbf{K}$ in graph $\mathcal{T}$, where $\mathbf{M}$ is (the known part of) the matching.

The first phase of adding vertices from $\mathbf{H}$ to $\mathbf{K}$ is called *adding by kernel neighbours*. Let a set of neighbours of vertex v in graph $\mathcal{D}$ be $\mathbf{N}_{\mathcal{D}}(v) \subseteq \mathbf{V}$, where $v \in \mathbf{W}$. We proceed as follows:

(i) Vertex $h \in \mathbf{H}$ is a member of the subset $\mathbf{H}' \subseteq \mathbf{H}$ if h has at least two kernel neighbours in graph $\mathcal{D}$, i.e.,

$$h \in \mathbf{H}' \quad \text{if} \quad |\mathbf{N}_K(h)| \geq 2,$$

where $\mathbf{N}_K$ are neighbours of $v \in V$ in $\mathbf{K}$, i.e., $\mathbf{N}_K(v) = \mathbf{N}_{\mathcal{D}}(v) \cap \mathbf{K}$.

(ii) Subsets $\mathbf{W}'(h) \subseteq \mathbf{W}$ are selected from $\mathbf{W}$ for every $h \in \mathbf{H}'$ as follows: Let $h \in \mathbf{H}'$ be the current candidate, $\mathbf{N}_K$ be the kernel neighbours of h in $\mathcal{D}$ and $\mathbf{M}(\mathbf{N}_K)$ be the corresponding vertices of $\mathbf{N}_K$ in $\mathcal{T}$. If a vertex $w' \in \mathbf{W}$ has at least two neighbours in $\mathbf{M}(\mathbf{N}_K)$, then $w' \in \mathbf{W}'(h)$. This way, we get $\mathbf{W}'(h)$ for each of $h \in \mathbf{H}'$, see Figure 4.4(a). Formally,

$$w' \in \mathbf{W}'(h) \quad \text{if} \quad |\mathbf{N}_{\mathcal{T}}(w') \cap \mathbf{M}(\mathbf{N}_K)| \geq 2,$$

where $\mathbf{N}_K(v) = \mathbf{N}_{\mathcal{D}}(v) \cap \mathbf{K}$.

(iii) Subset $\mathbf{W}''(h) \subseteq \mathbf{W}'(h)$ is selected from $\mathbf{W}'(h)$ if definite bitcode of h agrees with the bitcode in the template. Formally,

$$w'' \in \mathbf{W}''(h) \quad \text{if} \quad b(h) \wedge m = b(w'') \wedge m,$$

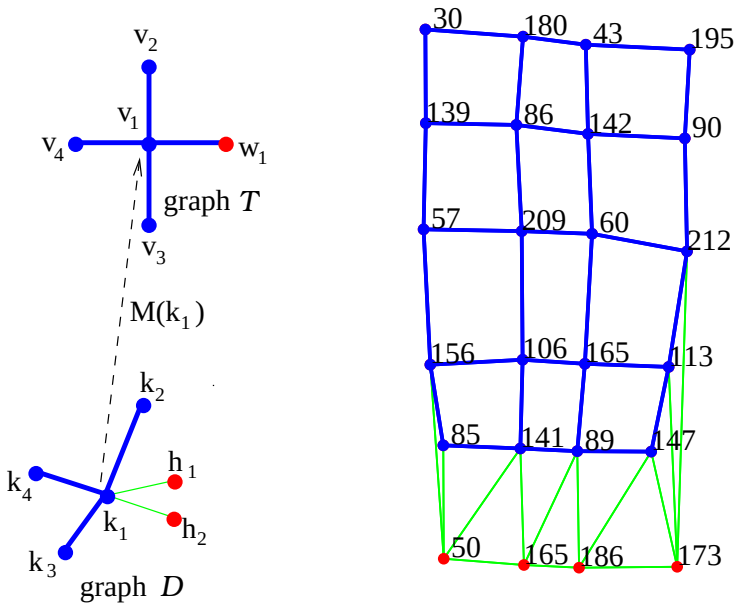
where $b(h)$ is the bitcode of h, $\wedge$ is a bitwise AND operation and m is a mask, that defines if a bit of the bitcode is definite or indefinite,

$$m_i = \begin{cases} 1, & \text{definite bit} \\ 0, & \text{otherwise} \end{cases}, i = 1, \ldots, 8.$$

This way, we get pre-matching $\bar{\mathbf{M}}(h) = \mathbf{W}''(h)$ for every vertex $h \in \mathbf{H}'$. Note that $|\bar{\mathbf{M}}|$ can be larger than unity. The *epipolar pruning* is the task to find a correct $w \in \mathbf{W}''(h)$ for current h. For every $h \in \mathbf{H}'$ and its corresponding vertex $w \in \mathbf{W}''$ a vertex $z_i = (\mathbf{M}^i)^{-1}(w)$ is found in each camera i. If every z_i satisfies the epipolar constraint with h then $w \in \hat{\mathbf{W}}$. If the number of vertices in $\hat{\mathbf{W}}$ is unity i.e. if $\hat{\mathbf{W}} = \{w\}$, then the bitcode $b(h)$ is replaced by $b(w)$, h is removed from the hull $\mathbf{H}$ and is added to the kernel $\mathbf{K}$.

Figure 4.4(a) illustrates the mapping $\mathbf{M}$ from $\mathbf{K}$ to $\mathbf{W}$. Kernel vertices in $\mathcal{D}$ and their images in $\mathcal{T}$ are drawn by blue colour. Vertex $h \in \mathbf{H}'$ and its set of correspondence candidates in template $\mathbf{W}'(h) \subseteq \mathbf{W}$ is shown by red colour.[1]

[1]The green edges in $\mathcal{D}$ are edges from $\mathbf{N}_K = \mathbf{N}_{\mathcal{D}}(v) \cap \mathbf{K}$.



(a) neco2 (b) neco1

Figure 4.5: Propagation of kernel vertices by known configuration

Figure 4.4(b) illustrates a real instance of graph $\mathcal{D}$. The subgraph $\mathcal{K}$ of $\mathcal{D}$ induced by $\mathbf{K}$ is shown by blue colour. The vertices 170, 120, and 202 are members of set $\mathbf{H}' = \{171, 104, 200\}$ because they have at least two neighbours in $\mathbf{K}$. For vertex $h = 171$, kernel neighbours are $\mathbf{N}_k(h) = \{27, 77\}$ and a set $\mathbf{W}'(h)$ taken from template $\mathcal{T}$ is $\mathbf{W}'(h) = \{170, 120\}$.

The second phase of adding vertices from $\mathbf{H}$ to $\mathbf{K}$ is called *adding by kernel vertex propagation*. Let graph $\mathcal{K} = (\mathbf{K}, \mathbf{E}_{DK})$ be a subgraph of $\mathcal{D}$ induced by $\mathbf{K}$. Every vertex $k \in \mathbf{K}$ is in correspondence with some $w \in \mathbf{W}$ such that $w \in \mathbf{M}(k)$.

(i) Let $\mathbf{K}'$ be a subset of $\mathbf{K}' \subseteq \mathbf{K}$ such that each has by one neighbour less in $\mathcal{K}$ than in $\mathcal{T}$:

$$k \in \mathbf{K}' \quad \text{if} \quad |\mathbf{N}_{\mathcal{K}}(k)| = |\mathbf{N}_{\mathcal{T}}(w)| - 1, \quad \{w\} = \mathbf{M}(k).$$

(ii) Those vertices $\mathbf{K}'' \subseteq \mathbf{K}'$ are selected from $\mathbf{K}'$, that have any neighbour in $\mathbf{H}$.[2]

$$k \in \mathbf{K}'' \quad \text{if} \quad |\mathbf{N}_H(k)| \geq 1,$$

where $\mathbf{N}_H$ are neighbours of $v \in \mathbf{V}$ in $\mathbf{H}$, i.e. $\mathbf{N}_H(v) = \mathbf{N}_{\mathcal{D}}(v) \cap \mathbf{H}$.

[2]This means, they have a missing neighbour in $\mathcal{D}$.

(iii) Let for a vertex $k \in \mathbf{K}''$ the vertex $w \in \mathbf{W}$ be the neighbour of a vertex $\mathbf{M}(k)$ in $\mathcal{T}$, that have no correspondence in $\mathbf{K}''$.[3] A vertex w is a member of $w \in \mathbf{W}'(k)$. Formally,

$$\mathbf{W}'(k) = \mathbf{N}_{\mathcal{T}}(\mathbf{M}(k)) \setminus \mathbf{M}(\mathbf{N}_K),$$

where $w \in \mathbf{W}'$ and $\mathbf{N}_K = \mathbf{N}_{\mathcal{D}}(k) \cap \mathbf{K}$. This way for every $k \in \mathbf{K}$ we get $\mathbf{W}'(k)$, where $0 \leq |\mathbf{W}'(k)| \leq 1$.

(iv) Each $h \in \mathbf{N}_D(k) \cap \mathbf{H}$ receives the set $\mathbf{W}'(k)$ as candidate matches in $\mathcal{T}$. We will denote the set $\mathbf{W}'(k)$ as the set $\mathbf{W}'(h)$.

(v) A subset $\mathbf{W}''(h) \subseteq \mathbf{W}'(h)$ is selected for every vertex $h \in \mathbf{H}$. A vertex $w \in \mathbf{W}''$ and $w \in \mathbf{W}'$ if definite bits of h is consistent with bits of w.

$$b(h) \wedge m = b(w) \wedge m,$$

where m is a mask as before.

(vi) Correct vertex from $\mathbf{W}''(h)$ is further reduced by epipolar pruning as described in adding by kernel neighbours.

Figure 4.5(a) illustrates the mapping of vertex k_1 in graph $\mathcal{D}$ to v_1 in graph $\mathcal{T}$ by $\mathbf{M}(k_1)$. The neighbours of k_1 in $\mathcal{D}$ are $\mathbf{N}_{\mathcal{K}} = \{k_2, k_3, k_4\}$ and the neighbours of v_1 in $\mathcal{T}$ are $\mathbf{N}_{\mathcal{T}} = \{v_2, v_3, v_4, w_1\}$, thus $\mathbf{W}'(k_1) = \{w_1\}$ and $\mathbf{W}''(h_1) = \mathbf{W}''(h_2) = \{w_1\}$.

Figure 4.5(b) illustrates a real instance of kernel vertex propagation. Vertices 85, 141, 89 and 147 are kernel candidates for propagation $\mathbf{K}'' = \{85, 141, 89, 147\}$. For example, missing neighbour in $\mathcal{T}$ of vertex 141 is vertex $\mathbf{W}'(141) = \{39\}$. Hull neighbours of vertex 141 $\mathbf{N}_H(141) = \{50, 165\}$ receive its set $\mathbf{W}'$ such that $\mathbf{W}''(50) = \{39\}$ and $\mathbf{W}''(165) = \{39\}$. The correct one of them is selected by epipolar pruning.

Since the graph matching matches the correct neighbour of $\mathbf{K}$ in graph $\mathcal{D}$, Hamming distance at least four between neighbouring vertices in template $\mathcal{T}$ decreases probability of mismatch. In this case, the probability that two neighbouring vertices will have all definite bit the same is reduced. Hamming distances of neighbouring vertices are shown in Figure 4.1 in each edge.

4.2.3 Epipolar Matching

Epipolar matching is the last stage of graph matching. It is the task of joining the remaining hull vertices, which can not be joined by kernel neighbours or

[3] With respect to inverse mapping $\mathbf{M}^{-1}$.

by kernel propagation because there have no valid edges between them and the kernel vertices. Epipolar matching is based on the bitcode constraint and the epipolar constraint only. It proceeds as follows:

(i) For every vertex $h \in \mathbf{H}$ subsets $\mathbf{W}'(h) \subseteq \mathbf{W}$ are selected template vertices, which have bitcode consistent with h. Formally:

$$b(h) \wedge m = b(w) \wedge m.$$

This way we get for every vertex $h \in \mathbf{H}$ pre-matching $\bar{\mathbf{M}}(h) = \mathbf{W}'(h)$.

(ii) The correct one is found by epipolar pruning.

To summarise, our matching procedure finds corresponding marks by first using the reliably decoded bitcodes, then by neighbourhood validation to reliably decoded marks, and finally by epipolar matching. This is unlike in the ELITE [8], [2], which requires the marks to move within a fixed region of space only. The constraint of our system is that each mark to be detected must be visible by at least two cameras under viewing angle not exceeding about 70° (0° corresponds to perpendicular viewing).

Chapter 5

Experimental Results

5.1 Calibration Error

We would like to establish measuring error for the calibration process. It is related to accuracy of estimation of projection matrix $\mathbf{P}_i$ for each camera i. The process of computing $\mathbf{P}_i$ and the calibration of measuring system is described in Section 2.2. The $\mathbf{P}_i$ can be computed correctly (except for numerical error) if the positions of corresponding points are known accurately. In this case, the most important source of error is caused by the calibration point detection error in calibration images and the accuracy of calibration panel and its movement in the z coordinate.

The experiment for measuring geometric error was done for front and back coordinate systems separately, as follows:

(i) Four sets of calibration figures were grabbed for positions 0mm, 60mm, and 120mm of the calibration panel by repeating the calibration data acquisition procedure. This way we get four independent calibrations $k = 1, \ldots, 4$.

(ii) The calibration points $\mathbf{x}_i^k$ were detected from calibration images and the projection matrices $\mathbf{P}_i^k$ were computed from $\mathbf{x}_i^k$ for each of camera i and for each of calibration $k = 1, \ldots, 4$.

(iii) For every pair of $\mathbf{x}_i^m$ and $\mathbf{P}_i^n$ points $\mathbf{X}^{m\,n}$ were reconstructed, where $m = 1, \ldots, 4$ and $n = 1, \ldots, 4$.

The mean reconstruction errors for the front cameras are shown in Table 5.1 and for the back cameras are shown in Table 5.2. Every element in the tables is the mean reconstruction error computed by

$$\epsilon^{m\,n} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{X}_i^{m\,n} - \mathbf{X}_i\| \quad \text{for} \quad m = 1, \ldots, 4, \quad n = 1, \ldots, 4, \tag{5.1}$$

[mm]	$\mathbf{P}_i^1$	$\mathbf{P}_i^2$	$\mathbf{P}_i^3$	$\mathbf{P}_i^4$
$\mathbf{x}_i^1$	0.0830	0.0838	0.0910	0.0897
$\mathbf{x}_i^2$	0.0896	0.0841	0.1020	0.0958
$\mathbf{x}_i^3$	0.0870	0.0951	0.0833	0.0887
$\mathbf{x}_i^4$	0.0914	0.0922	0.0952	0.0860

Table 5.1: Mean reconstruction errors for the front cameras in millimetres

[mm]	$\mathbf{P}_i^1$	$\mathbf{P}_i^2$	$\mathbf{P}_i^3$	$\mathbf{P}_i^4$
$\mathbf{x}_i^1$	0.1502	0.1512	0.1497	0.1513
$\mathbf{x}_i^2$	0.1558	0.1460	0.1534	0.1541
$\mathbf{x}_i^3$	0.1503	0.1499	0.1473	0.1530
$\mathbf{x}_i^4$	0.1526	0.1500	0.1530	0.1487

Table 5.2: Mean reconstruction errors for the back cameras in millimetres

where points $\mathbf{X}^{m\,n}$ are reconstructed from image points $\mathbf{x}_i^m$ by projection matrices $\mathbf{P}_i^n$, $\mathbf{X}$ are their true positions, and N is the total number of points $\mathbf{X}^{m\,n}$. The standard deviations s for elements $\epsilon^{m\,n}$ in the Tables 5.1 and 5.2 are computed by

$$s = \sqrt{\frac{1}{16}\sum_{m=1}^{4}\sum_{n=1}^{4}\left(\epsilon^{m\,n} - \bar{\epsilon}\right)^2}, \quad \text{where} \quad \bar{\epsilon} = \frac{1}{16}\sum_{m=1}^{4}\sum_{n=1}^{4}\epsilon^{m\,n}. \tag{5.2}$$

The standard deviation was 0.0054 for the front cameras and it was 0.0026 for the back cameras.

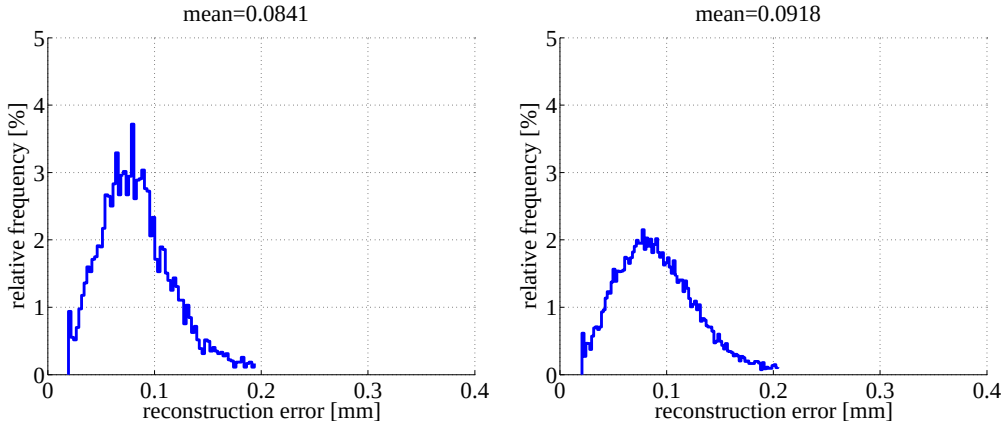
The histograms of geometric errors are computed for front and back cameras separately and are shown in Figure 5.1. The histograms shown in Figures 5.1(a) and 5.1(c) illustrate the mutual residuals errors. Their spatial geometric errors ϵ_i are computed by

$$\epsilon_i = \|\mathbf{X}_i^{m\,m} - \mathbf{X}_i\|, \quad \text{for} \quad m = 1, \ldots, 4. \tag{5.3}$$

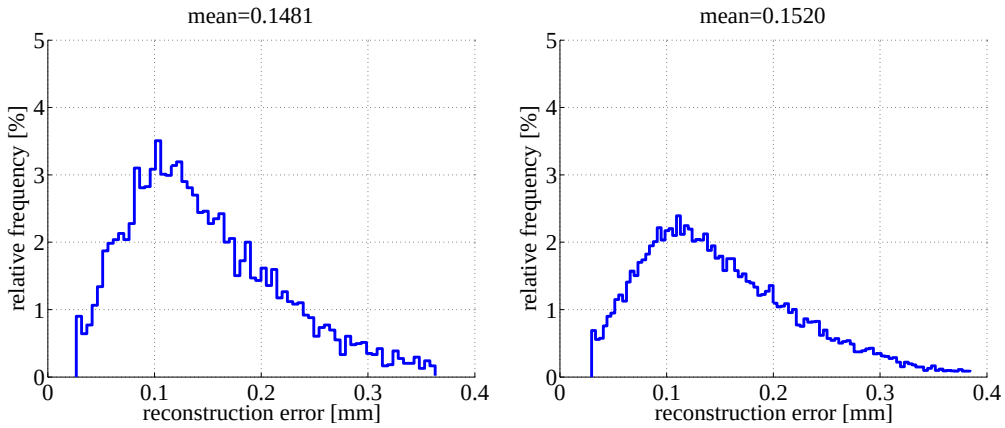
The estimation of the low error rate is computed by $\frac{1}{N}\sum_{i=1}^{N}\epsilon_i$, where ϵ_i are the residuals geometric errors taken from (5.3). The low error rate of the calibration was 0.0841mm for the front cameras and it was 0.1481mm for the back cameras. The histograms shown in Figures 5.1(b) and 5.1(d) illustrate the independent spatial geometric errors ϵ_i computed by

$$\epsilon_i = \|\mathbf{X}_i^{m\,n} - \mathbf{X}_i\|, \quad \text{for} \quad m = 1, \ldots, 4, \quad m = 1, \ldots, 4, \tag{5.4}$$

where $m \neq n$. The estimation of the high error rate is computed by $\frac{1}{N}\sum_{i=1}^{N}\epsilon_i$, where ϵ_i are the spatial geometric errors taken from (5.4). The high error rate



(a) residual errors for the front cameras (b) independent errors for the front cameras



(c) residual error for the back cameras (d) independent errors for the back cameras

Figure 5.1: Histograms of calibration errors in millimetres for the front and the back cameras, respectively

of the calibration was 0.0918mm for the front cameras and it was 0.1520mm for the back cameras.

The true error rate of calibration is between values $0.0841 \div 0.0918$mm for the front cameras and $0.1481 \div 0.1520$mm for the back cameras. The mean reconstruction errors in the front cameras are smaller then in the back cameras because the points $\mathbf{X}^{m\,n}$ are reconstructed from four corresponding points $\mathbf{x}_i^m$ in front space but in the back space by two only.

5.2 Calibration Repeatability

Repeatability of calibration is based on the results from the measurement of calibration error. In this experiment the non-flatness error of calibration panel

[mm]	$\mathbf{X}_{m,1}$	$\mathbf{X}_{m,2}$	$\mathbf{X}_{m,3}$	$\mathbf{X}_{m,4}$
$\mathbf{X}_{1,n}$	0	0.0236	0.0135	0.0189
$\mathbf{X}_{2,n}$	0.0236	0	0.0181	0.0219
$\mathbf{X}_{3,n}$	0.0135	0.0181	0	0.0187
$\mathbf{X}_{4,n}$	0.0189	0.0219	0.0187	0

Table 5.3: Mean residual errors for the front cameras in millimetres

[mm]	$\mathbf{X}_{m,1}$	$\mathbf{X}_{m,2}$	$\mathbf{X}_{m,3}$	$\mathbf{X}_{m,4}$
$\mathbf{X}_{1,n}$	0	0.0571	0.0513	0.0423
$\mathbf{X}_{2,n}$	0.0571	0	0.0460	0.0530
$\mathbf{X}_{3,n}$	0.0513	0.0460	0	0.0458
$\mathbf{X}_{4,n}$	0.0423	0.0530	0.0458	0

Table 5.4: Mean residual errors for the back cameras in millimetres

is ignored.

The results of repeatability of calibration for the front cameras are shown in Table 5.3 and for the back cameras are shown in Table 5.4. Every element in the tables are computed by

$$\epsilon^{m\,n} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{X}^m - \mathbf{X}^n\| \quad \text{for} \quad m = 1,\ldots,4, \quad n = 1,\ldots,4, \tag{5.5}$$

where the $\mathbf{X}^m$ and $\mathbf{X}^n$ are reconstructed points from detected calibration points $\mathbf{x}_i^m$ and $\mathbf{x}_i^n$ from camera i by projection matrices $\mathbf{P}_i^m$ and $\mathbf{P}_i^n$ and N is the total number of points $\mathbf{X}_i^m$ and $\mathbf{X}_i^n$.

The histograms shown in Figure 5.2 illustrate all spatial geometric errors for the front and the back cameras separately, which are computed by

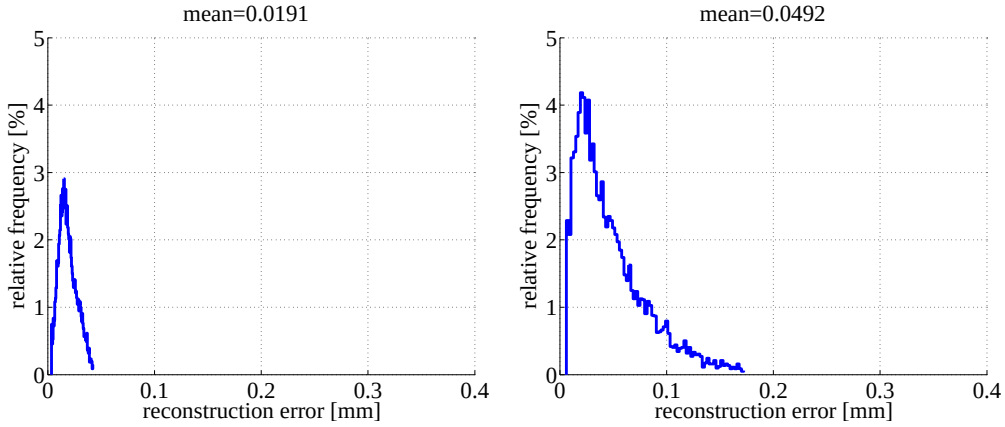
$$\|\mathbf{X}_i^m - \mathbf{X}_i^n\|, \quad \text{for} \quad m = 1,\ldots,4 \quad \text{and} \quad n = 1,\ldots,4, \tag{5.6}$$

where $m \neq n$.

From Tables 5.3 and 5.4 it is obvious, that the repeatability is smaller than calibration error. Thus, the calibration error is more influenced by non-flatness calibration object than the calibration point detection error.

5.3 Panda Mark Detection Repeatability

The measuring of repeatability of Panda mark detection is a very important experiment. It determines the measuring accuracy of Panda marks on human



(a) for the front cameras (b) for the back cameras

Figure 5.2: Repeatability of calibration

[mm]	$\mathbf{X}_{m,1}$	$\mathbf{X}_{m,2}$	$\mathbf{X}_{m,3}$	$\mathbf{X}_{m,4}$	$\mathbf{X}_{m,5}$	$\mathbf{X}_{m,6}$
$\mathbf{X}_{1,n}$	0	0.1056	0.0609	0.0841	0.1112	0.0759
$\mathbf{X}_{2,n}$	0.1056	0	0.0885	0.0806	0.0934	0.0920
$\mathbf{X}_{3,n}$	0.0609	0.0885	0	0.0588	0.0756	0.0664
$\mathbf{X}_{4,n}$	0.0841	0.0806	0.0588	0	0.0988	0.0695
$\mathbf{X}_{5,n}$	0.1112	0.0934	0.0756	0.0988	0	0.0735
$\mathbf{X}_{6,n}$	0.0759	0.0920	0.0664	0.0695	0.0735	0

Table 5.5: Mean residual errors for the back cameras in millimetres

trunk. This error includes movement errors of the calibration panel in the z coordinate and Panda mark detection error in images.

The experiment was done as follows:

(i) Calibration images were acquired and the projection matrices $\mathbf{P}_i$ were computed for every camera i by our standard calibration procedure.

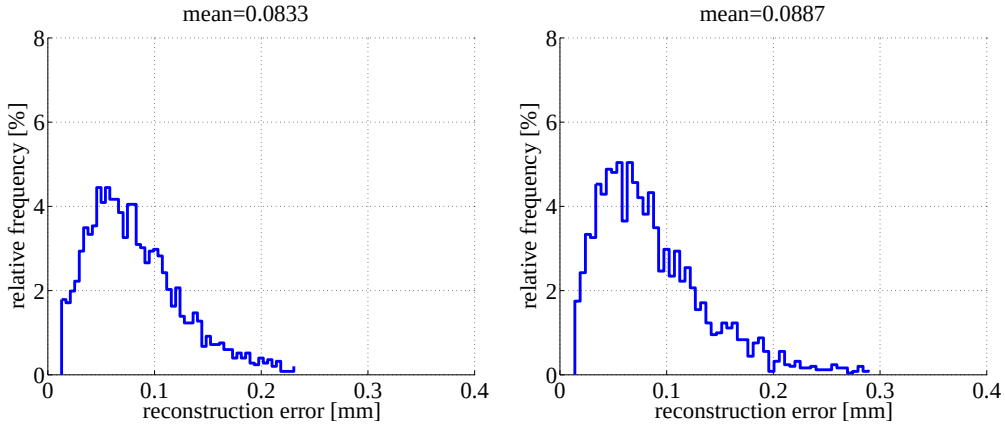
(ii) A sheet of Panda marks were placed on the calibration panel and the images were grabbed in same positions of the calibration panel[1] as in measuring the calibration error. This was repeated six times.

(iii) For every measurement $j = 1, \ldots, 6$, the positions of Panda marks $\mathbf{x}_i^j$ were detected in images and their 3D positions $\mathbf{X}^j$ were reconstructed by projection matrices $\mathbf{P}_i^j$, where i is the number of camera.

[1]The panel was grabbed in positions 0mm, 60mm, and 120mm.

[mm]	$\mathbf{X}_{m,1}$	$\mathbf{X}_{m,2}$	$\mathbf{X}_{m,3}$	$\mathbf{X}_{m,4}$	$\mathbf{X}_{m,5}$	$\mathbf{X}_{m,6}$
$\mathbf{X}_{1,n}$	0	0.0848	0.0955	0.0872	0.0897	0.0805
$\mathbf{X}_{2,n}$	0.0848	0	0.0836	0.0885	0.0887	0.0966
$\mathbf{X}_{3,n}$	0.0955	0.0836	0	0.0870	0.0946	0.0921
$\mathbf{X}_{4,n}$	0.0872	0.0885	0.0870	0	0.1032	0.0700
$\mathbf{X}_{5,n}$	0.0897	0.0887	0.0946	0.1032	0	0.0729
$\mathbf{X}_{6,n}$	0.0805	0.0966	0.0921	0.0700	0.0729	0

Table 5.6: Mean residual errors for the back cameras in millimetres



(a) for the front cameras (b) for the back cameras

Figure 5.3: Repeatability of Panda mark detection

The results of repeatability errors are shown in Table 5.5 for the front cameras and they are shown in Table 5.6 for the back cameras. Elements in the tables are computed as mean values of spatial distances of reconstructed points taken from the same mark. Formally,

$$\epsilon^{m\,n} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{X}_i^m - \mathbf{X}_i^n\| \quad \text{for} \quad m = 1,\ldots,6, \quad n = 1,\ldots,6, \tag{5.7}$$

where the $\mathbf{X}^m$ and $\mathbf{X}^n$ are reconstructed points from detected Panda marks $\mathbf{x}_i^m$ and $\mathbf{x}_i^n$ from camera i by projection matrices $\mathbf{P}_i^m$ and $\mathbf{P}_i^n$ and N is the total number of points $\mathbf{X}_i^m$ and $\mathbf{X}_i^n$.

The histograms shown in Figure 5.3 illustrate all geometric errors ϵ_i for front and back cameras separately, which are computed by

$$\epsilon_i = \|\mathbf{X}_i^m - \mathbf{X}_i^n\| \quad \text{for} \quad m = 1,\ldots,6, \quad n = 1,\ldots,6, \tag{5.8}$$

where $m \neq n$.

The goal for this experiment is to find the conditions under which Panda mark detection fails. In this experiment we focus on distance from cameras as the main cause of the effect.

5.4 Failure Limits of Panda Mark Detection

The experiment of Panda mark detection failure was done for the back cameras only, as follows:

(i) The system was calibrated by the standard calibration procedure (as described in Section 6.2).

(ii) Front and back Panda marks were printed on a sheet of paper in configuration required by the template (see Figure 4.1).

(iii) The sheet of Panda marks was placed on the calibration panel.

(iv) Calibration panel was moved in the z direction by 50mm steps and a pair of images were grabbed in each position.

(v) Images of Panda marks were processed.

Failure limits of Panda marks is shown in Table 5.7, where p is the number of the panel position, d_1 and d_2 are distances between a centre of gravity of reconstructed Panda marks and the centres of projection of the back cameras for each camera (d_1 for the back camera 5 and d_2 for the back camera 6), m is a number of reconstructed (matched) Panda marks (the maximum is 24), sd_1 and sd_2 are the numbers of successfully detected Panda marks with all definite bits in their respective bitcodes, dt_1 and dt_2 are total number of detected Panda marks (irrespective of the definiteness of their bitcode), and b_1 and b_2 are the total number of indefinite bits of all Panda mark bitcodes combined.

For positions, where no Panda mark was reconstructed, the positions were measured by a ruler (positions from 8 onward). Under standard working conditions in this project the distances are $d_1 = 1000 \div 1150$mm and $d_2 = 970 \div 1030$mm.

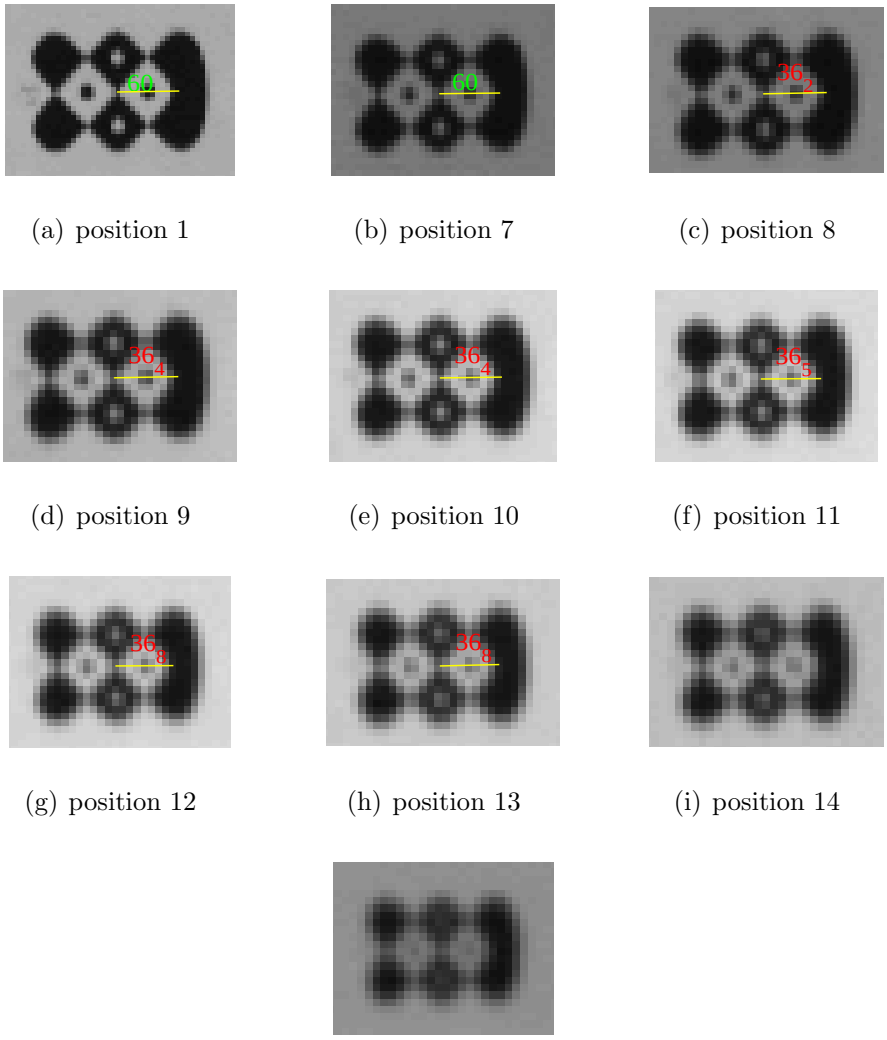
In Figure 5.4 a few examples of detected Panda mark in different positions (corresponding to Table 5.7) are shown. In this case, the Panda mark with bitcode 60 was selected. The image of Panda mark in the first (standard) distance of a panel is shown in Figure 5.4(a). The Figures 5.4(b) illustrates the last position, where the Panda mark was successfully detected. Figure 5.4(c) illustrates the position $p = 8$, where the Panda mark was detected with two indefinite bits. Figures 5.4(d) and 5.4(e) illustrate the positions $p = 9$ and $p = 10$, where the Panda mark was detected with four indefinite bits. Figure 5.4(f) illustrates the position $p = 11$, where the Panda mark was detected with

p	d_1 [mm]	d_2 [mm]	m	sd_1	sd_2	dt_1	dt_2	b_1	b_2
1	1100	1028	24	24	24	24	24	0	0
2	1151	1083	24	24	24	24	24	0	0
3	1198	1131	24	24	24	24	24	0	0
4	1242	1178	24	24	24	24	24	0	0
5	1293	1231	24	23	23	24	24	1	1
6	1341	1282	24	15	21	24	24	15	6
7	1390	1333	23	8	2	24	24	33	45
8	~ 1430	~ 1370	0	0	0	24	24	62	79
9	~ 1475	~ 1420	0	0	0	24	24	91	96
10	~ 1523	~ 1470	0	0	0	24	24	96	96
11	~ 1572	~ 1520	0	0	0	24	24	106	105
12	~ 1621	~ 1570	0	0	0	23	24	136	174
13	~ 1669	~ 1620	0	0	0	22	18	152	131
14	~ 1718	~ 1670	0	0	0	14	5	94	27
15	~ 1767	~ 1720	0	0	0	7	0	46	0
16	~ 1816	~ 1770	0	0	0	3	0	15	0
17	~ 1865	~ 1820	0	0	0	0	1	0	4
18	~ 1889	~ 1845	0	0	0	1	0	4	0

Table 5.7: Failure limits of Panda marks detection

five indefinite bits. Figures 5.4(g) and 5.4(h) illustrate the positions $p = 12$ and $p = 13$, where the Panda mark was detected with all eight indefinite bits. Figure 5.4(i) illustrates the first position $p = 14$, where the Panda mark was not detected. The image of Panda mark in the last position $p = 18$ is shown in Figure 5.4(j).

From Figures 5.4 it is obvious, that illumination was variable in the measuring positions p but this does not influence Panda mark detection unless intensities of pixels are not in saturation or close to zero. During this experiment every detected definite bits of Panda marks were valid. This is very important property of Panda mark detection because every false positive (wrong detected definite bits) are removed from solution by graph matching.



(a) position 1 (b) position 7 (c) position 8

(d) position 9 (e) position 10 (f) position 11

(g) position 12 (h) position 13 (i) position 14

(j) position 18

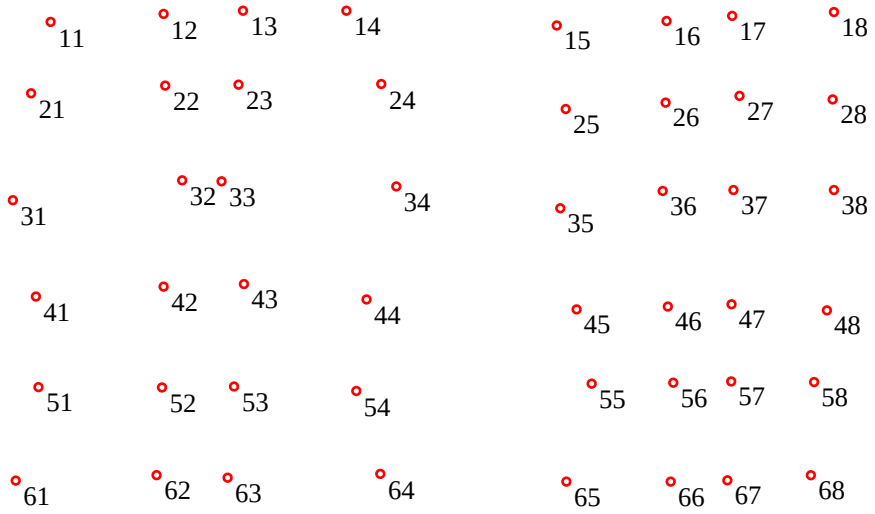
Figure 5.4: Results of Panda mark detection

Chapter 6

The Measurement Protocol

A measurement protocol will be describe in the following text. Roles such as the object, a medical specialist, and a technical operator participate in each measuring procedure.

6.1 Panda Mark Placement



(a) sternal side of trunk (b) dorsal side of trunk

Figure 6.1: Panda marks replacement on human trunk, projection to latero-lateral plane

We suppose that the Panda marks are printed on a sheet of stickers. A generation of Encapsulated PostScript file with them is described in Section 3. The Panda marks are placed on sternal and dorsal side of human trunk surface

in six rows and four columns. The projections of deployed Panda marks to latero-lateral planes are shown in Figure 6.1. Figure 6.1(a) illustrates the sternal side view and Figure 6.1(b) illustrates the dorsal side view.

At the beginning the Panda marks are placed in two middle columns[1] to the points of interest (i.e. ribs, parts of abdomen, and vertebral column) by the medical specialist. The remaining Panda marks are placed with the help of a projected laser lane. Since the stickers are so large (38mm by 21mm) and the measuring subject is not fixed during stickers replacement, it is complicated to find the same places of Panda marks repeatedly. Finding the mark placement is in progress actually.

6.2 Image Acquisition

Data are acquired via a gaphical user interface running under Matlab. It is processed under Windows operating system in Matlab 6.5.

During the Panda mark replacement the technical operator can perform initialization of frame grabbers by command `bopen`. This operation takes up to 5 minutes. Cameras can be focused by GUI `bfocus`, it shows live movie from current camera in full active resolution. The name of current measuring session is set by command `bsession`.

The following sequence of commands describes the real instance of measurement, where the name of the session is `experiment1`.

```
bopen                 %initialization
bfocus 1              %can be used for focalization of camera 1
bsession experiment1  %setting of current session
```

From this time, the measuring system is prepared for obtaining calibration images and images of the subject breathing movements of subject.

The process of *calibration image acquisition* is started by command `bcalib`. The following steps follow:

(i) Images of calibration panel are grabbed in positions 0mm, 60mm, and 120mm.

(ii) The calibration object is moved to homing position (toward the front cameras) such that it is out of vision angles of the front cameras. The calibration panel, which is visible in the rear cameras is covered by a sheet of the thick paper.

[1] At the beginning the front Panda marks with labels 12, 22, 32, 42, 52, 62, 13, 23, 33, 43, 53, 63 and back Panda marks with labels 16, 26, 36, 46, 56, 66, 17, 27, 37, 37, 47, 57, 67 are placed on human trunk.

(iii) The registration sheet is grabbed in three different arbitrary positions such that it is in view in all cameras.[2]

The process of *subject image acquisition* is started by command `blive` with the name of the measurement as a optional parameter. If the parameter is not specified then the measurement name is set to the default value. The image acquisition is done as follows:

(i) The support is raised and lowered to bring the subject to view in all cameras.

(ii) Images at inspiration and expiration phases of the measured subject are grabbed repeatedly. The inspiration is measured first. The time instants of maximum inspiration and maximum expiration are determined by the medical specialist.

(iii) Several types of breathing can be measured for subject:

  (a) normal breathing
  (b) deep breathing
  (c) breathing immediately after physical exercise

(iv) After every measurement is taken, the data must be processed by technical operator for their validation.

All grabbed images are saved to a shared directory on Unix operating system for processing.

6.3 Data Processing

Scripts for camera calibration, Panda mark detection, and reconstruction 3D positions are designed for Matlab 6.5, that works under Unix operating system.

Data are processed by the following steps:

(i) Initialization of parameters is done by command `bprocinit`. This is done once, at the beginning of a session.

(ii) A name of current session is set by command `bprocsession`.

(iii) A calibration is processed by command `bproccalib` from shared calibration figures.

(iv) 3D positions from detected Panda marks are computed by command `bproc`.

[2]The registration sheet is a paper with camera calibration figure printed double-sided. Both sides are in geometric correspondence by register.

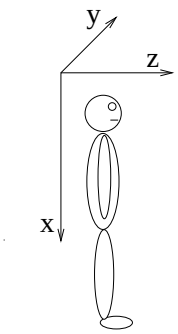
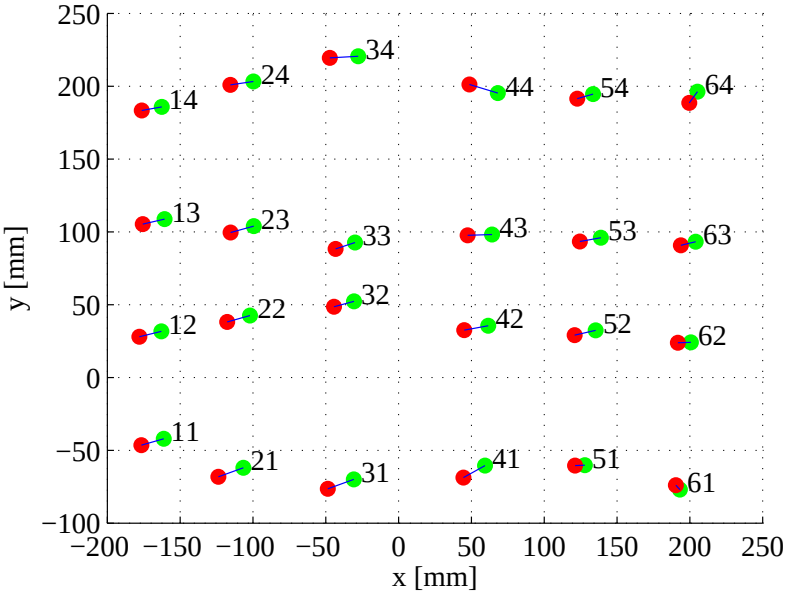


Figure 6.2: Coordinate system

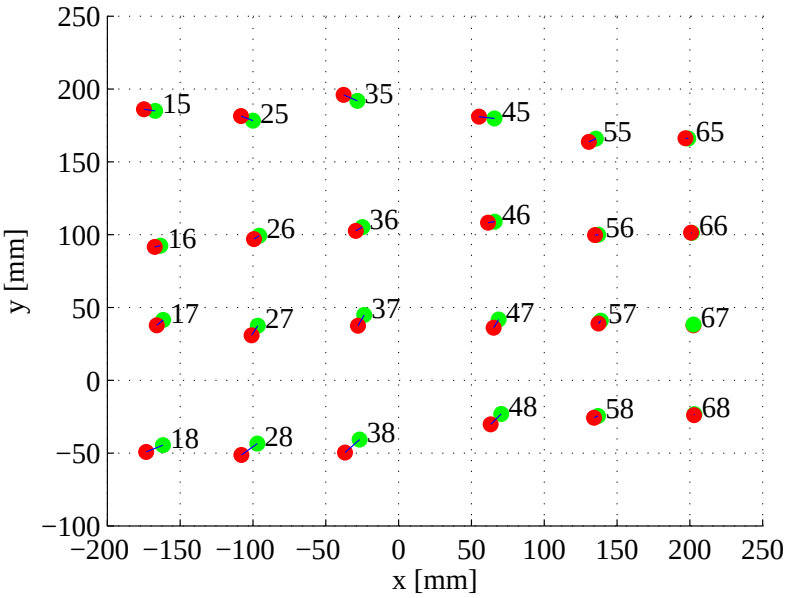
6.4 Data Visualization

Measured data are inspected in 3D coordinate system using a visualization GUI. It can be used for data validation after data processing stage. The visualisation GUI is started by command `bprocview`.

The reconstructed points can be projected to latero-lateral, transversal and sagittal plane or they can be shown in 3D view. An examples of the data reconstruction are shown in Figures 6.3, 6.4, 6.5 and 6.6. The position of marks inspiration is shown by red colour and the position of expiration is shown by green colour. The Figure 6.2 illustrates used coordinate system, where the x axis is vertical, oriented downward, the y axis is horizontal, oriented form the right side of the subject to his/her left side, and the z axis is the direction of the subject's line of sight.

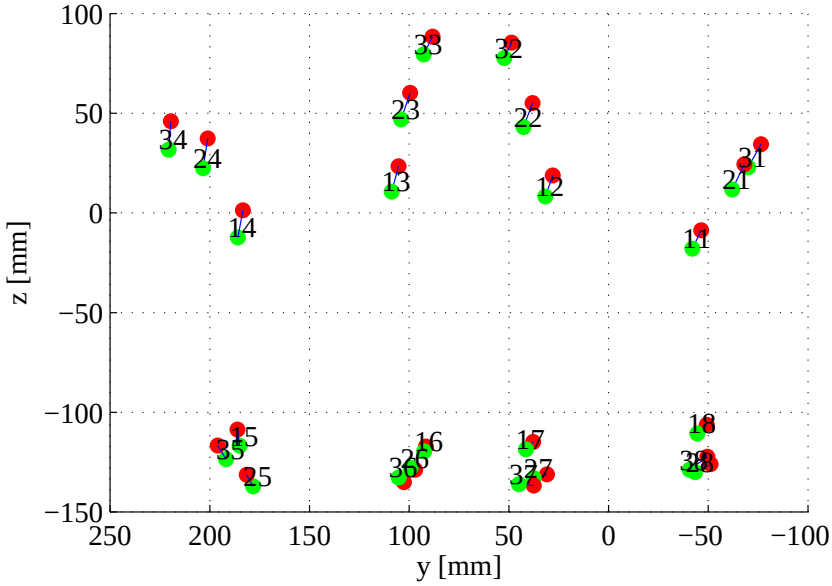


(a) front Panda marks

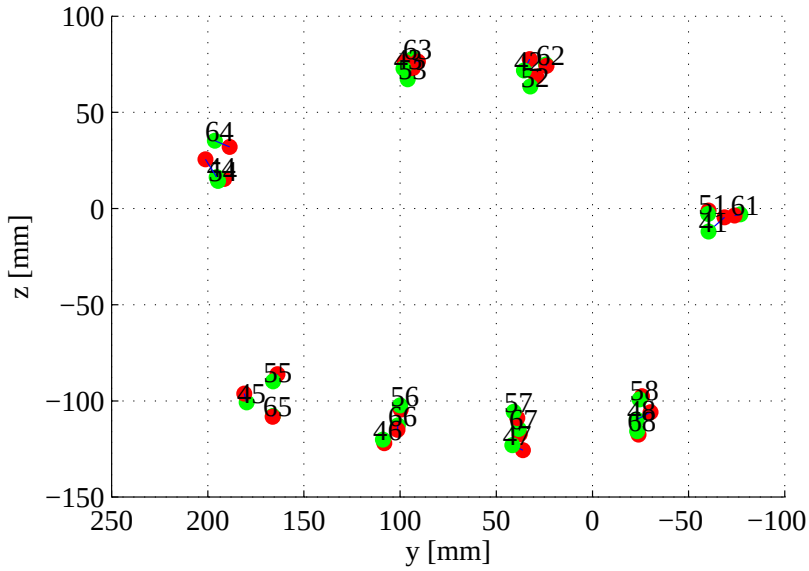


(b) back Panda marks

Figure 6.3: Projection to latero-lateral plane

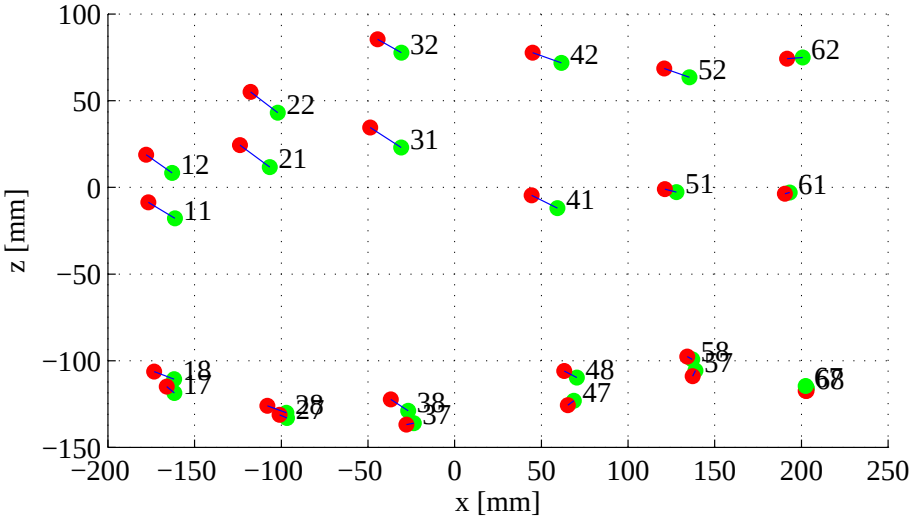


(a) top Panda marks



(b) bottom Panda marks

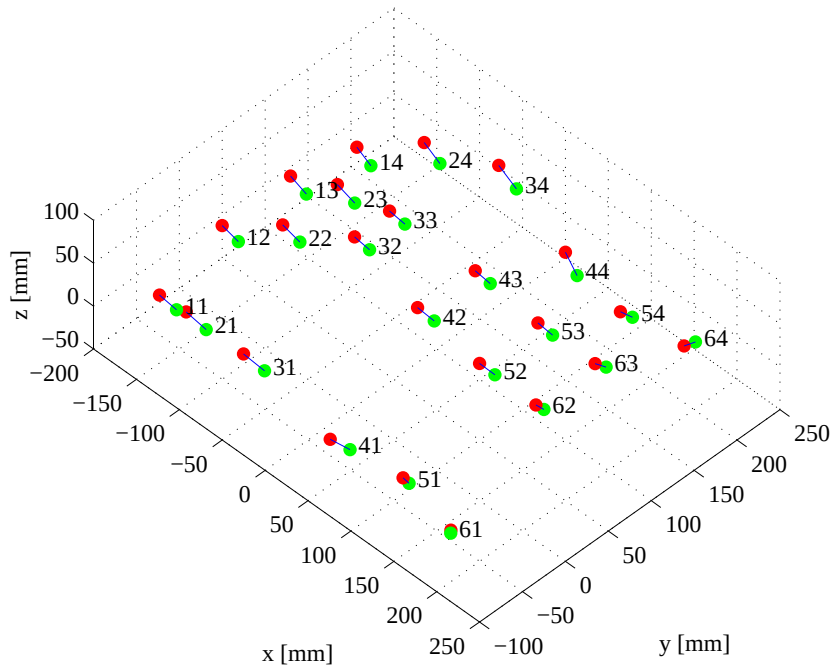
Figure 6.4: Projection to transversal plane



(a) left Panda marks

(b) right Panda marks

Figure 6.5: Projection to sagittal plane



(a) front Panda marks

(b) back Panda marks

Figure 6.6: 3D views

Chapter 7

Conclusions

The project is aimed to photogrammetric measurement of human breathing movements. The measuring system is able to obtain images of a subject, that has placed special marks on its trunk. The 3D positions of the marks are computed automatically from the images. This method was used for measuring ten subjects (one male and nine females). The experiments suggest that the measuring method is suitable for this purpose.

The next stage of this project will be quantification of measured data. The quantification must cope with following problems:

(i) repeatability our repeated data acquisition (the quantifier must not be sensitive to imprecisions in determining the position of maximal inspiration/expiration)

(ii) small sensitivity to non-breathing motions (rotations, translations, non-rigid motions)

(iii) small sensitivity to repeated mark replacement on the body

Since this is a three-year project, I participated in just several parts of its solution. I have derived SVD solution of the absolute orientation problem including correctness proof and degenerate case treatment and I have integrated it to the standard calibration procedure (as describes in Section 2.2.3). I have written major part of the GUI that is used to acquire data from the measurement system. I have designed and implemented the algorithm of graph matching, which is able to match poorly detected Panda marks (as described in Chapter 4). I also implemented the user interface for reconstruction and visualization results (see Section 6.4).

Bibliography

[1] Olivier Faugeras. *Three-Dimensional Computer Vision*. MIT Press, Cambridge, Massachusetts, 1993.

[2] G. Ferrigno, P. Carnevali, A. Aliverti, F. Molteni, G. Beulcke, and A Pedotti. Three-dimensional optical analysis of chest-wall motion. *Journal of Applied Physiology*, 77(3):1224–1231, September 1994.

[3] Gene H. Golub and Charles F. Van Loan. *Matrix Computation*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, USA, 3rd edition, 1996.

[4] Paul Richard Halmos. *Finite-dimensional Vector Spaces*. Undergraduate Texts in Mathematics. Springer, New York, USA, 2nd edition, 1987.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[6] John D. Hobby. *A User's Manual for MetaPost*. Murray Hill, 1995.

[7] 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 (Optics and Image Science)*, 5:1127–1135, July 1988.

[8] C. M. Kenyon, S. J. Cala, S. Yan, A. Aliverti, G. Scano, R. Duranti, A. Pedotti, and P.T. Macklem. Rib cage mechanics during quiet breathing and exercise in humans. *Journal of Applied Physiology*, 83(4):1242–1255, October 1997.

[9] Petr Mareček. Camera calibration systems. Master's thesis, Department of Cybernetics, Faculty of Electrical Engineering, Prague, May 2001. [In Czech].

[10] Radim Šára. Bundle adjustment in polynocular camera calibration. Department of Cybernetics, Faculty of Electrical Engineering in Prague, June 2004. Unpublished report.

[11] Radim Šára, Vladimír Smutný, Michaela Veverková, and Jiří Čumpelík. A photogrammetric method for measuring breathing movements. In Jiří Jan, Jiřík Kozumplík, and Ivo Provazník, editors, *Analysis of Biomedical Signals and Images, Proceedings of 16th Biennial International EURASIP Conference BIOSIGNAL 2002*, pages 329–331, Brno University of Technology, Brno, Czech Republic, June 2002. VUTIUM Press.

[12] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. A comprehensive survey of bundle adjustment in computer vision. In *Proc. Vision Algorithms: Theory and Practice. International Workshop on Vision Algorithms*, number 1883 in Lecture Notes in Computer Science, pages 298–372. Springer Verlag, 1999.

[13] Vít Zýka. Panda marks description and detection. Department of Cybernetics, Faculty of Electrical Engineering, September 2003. Unpublished report.