Upper body postural assessment for common dentistry visual aids

Master Thesis in Computer Science Engineering

Emilio Trotta, mat. 837134

Abstract

Postural assessment is a fundamental aspect for preventing long-term musculoskeletal disorders (MSDs) due to fatiguing jobs. Operative dentistry also belongs to this category and we developed a Computer Vision approach to automatically analyze the dentist posture during the operations,so obtaining an evaluation of MSD risks, performed accordingly to well established assessment criteria such as RULA (Rapid Upper Limb Assessment). Three different visualization set-ups were analyzed and compared from a postural point of view. The considered methods were: naked eyes, medical loupes and surgical microscope. The results show a significantly better assessment when the microscope is used and they validate our approach both as a feasible and effective method for postural assessment in fatiguing jobs and a continuous monitoring method of the job activity. The whole adopted procedure follows a non-invasive approach based on Augmented Reality markers tracked from a distant camera. This methodology can be extended for monitoring different working activities and providing an accurate postural evaluation in order to prevent MSDs.

Sommario

La valutazione della postura è un aspetto fondamentale per la prevenzione di disturbi muscolo-scheletrici (DMS) causati da attività faticose. Tra queste occupa un posto anche la chirurgia orale: nel nostro lavoro abbiamo sviluppato un metodo basato sulle tecniche della Computer Vision che permette di analizzare la postura del dentista durante le operazioni, in modo da ottenere una valutazione dei rischi di DMS; quest'ultima eseguita sulla base di metodi di valutazione consolidati,come ad esempio RULA. Tre diverse tecniche di visualizzazione sono state analizzate e messe a confronto, ovvero: occhio nudo, occhialetti ingrandenti e microscopio chirurgico. I risultati mostrano una valutazione decisamente più positiva per le operazioni svolte con il microscopio e convalidano il nostro metodo come applicabile ed efficace per l'osservazione e la valutazione posturale durante attività faticose. La procedura segue un approccio non invasivo basato sull'utilizzo di markers per la realtà aumentata tracciati da una telecamera. Questa metodologia può essere estesa al monitoraggio di altre attività con lo scopo di valutare la postura e prevenire eventuali DMS.

Contents

1
Introduction

In this written work we present a system for body posture analysis meant for monitoring the upper body posture of an operating dentist. The method used for data acquisition is based onto the operator position 3D reconstruction, particularly focusing on the upper body. The method implemented is completely non-invasive as it is based on a markers-equipped t-shirt to be worn by the subject and on two cameras. The goal of this system is that of obtaining a parametric 3D description of the monitored body by analyzing successive frames containing the image of the planar markers attached to the t-shirt and assessing the quality of the postures assumed during the operations.

The procedure presented in this thesis was requested and developed for the San Paolo Hospital of Milan, in collaboration with doctor Alberto Pispero, who's body posture was evaluated throughout the operations.

We collected the data for 60 surgical operations regarding the removal of 38 and 48 tooth, also known as wisdom tooth. The data was then analyzed following an ergonomics assessment method.

The research of a mathematical model for a real object in three dimensions from one or more images is a well known problem in the computer vision field and we want to present a valid approach which allows us to reconstruct a human body posture with few requirements in the image acquisition procedure.

The block diagram shown in figure 1.1 schematizes the different steps we followed for collecting and elaborating the data in order to achieve 3D reconstruction and the consequent posture assessment.

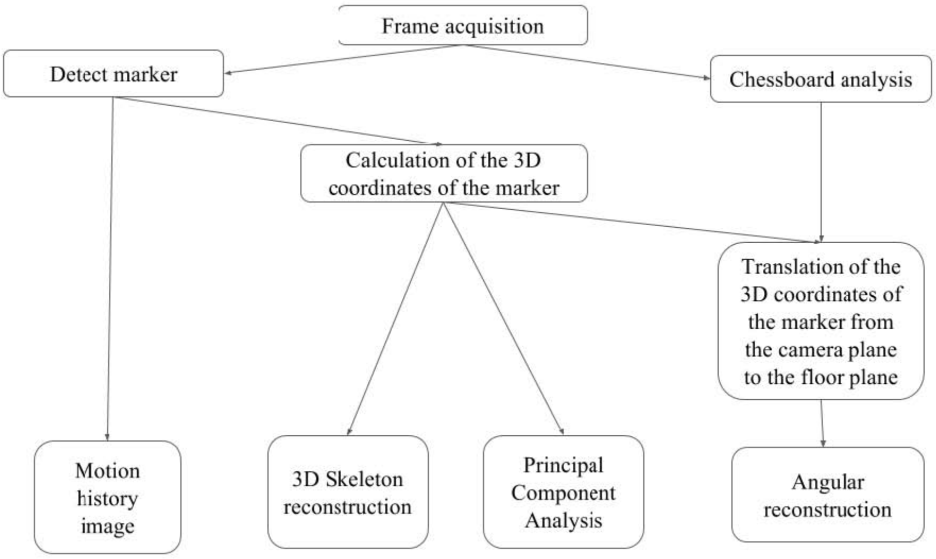

Frame acquisition
Detect marker
Chessboard analysis
Calculation of the 3D coordinates of the marker
Translation of the 3D coordinates of the marker from the camera plane to the floor plane
Motion history image
3D Skeleton reconstruction
Principal Component Analysis
Angular reconstruction

Figure 1.1: Block diagram for the posture evalutation procedure

This written work is structured in the following way:

In the second chapter the preliminary theoretical background is presented. In particular we introduce the mathematical tools and algorithms necessary to describe the basic elements of projection geometry, camera modeling and camera calibrations adopted in computer visions in order to estimate objects position in space.

In the third chapter we discuss about body posture assessment methodologies. We introduce some of the key elements necessary for evaluating the quality of a body posture and the postural aspects that can increment the risk for musculoskeletal disorders. Besides, in this chapter two of the most known postural assessment criteria are presented and described. In particular we discuss about *Rapid Upper Limb Assessment (RULA)*, adopted in our application, and *Novel Ergonomic Postural Assessment Method (NERPA)*.

In the fourth chapter we present the developed model for body posture estimation, the necessary equipment and setup and the methodology used in our data collection and analysis. In particular, in the first part we show the experimental setup and the arrangement of the operating room with a short description of the hardware used, in the second one we describe the training phase in which "teach" our model how to detect the markers. Then the third part is about the detection phase, with a the description of the software detection procedure. The fourth and final part is about the analysis phase.

In the fifth chapter we present the obtained experimental result processed through our analysis model. These results have been achieved from a sample of 60 dental operations: 20 performed with the aid of a microscope, 20 performed with the aid of dental loupes and 20 without using any magnification tool, just the naked eye. We consider and compare all the motion history images computed and the relevant angular values collected in order to contrast the different operating methods examined in our study. Finally, we use the elaborated data in order to perform an average postural assessment of the three approaches following RULA algorithm.

In the sixth and final chapter of the thesis we present the conclusions of our study.

2

Theoretical Background

This chapter will be devoted to present the mathematical tools and the basic algorithms necessary to contextualize this thesis project within the interdisciplinary field of Computer Vision.

2.1 Camera model

The camera is the instrument through which a 3D object in real space can be represented as a 2D object in the plane defined as *image plane.* The points belonging to the transformed 2D object are the projections of the 3D points forming the original object. The final result of this projection operation strongly depends on the technical characteristics of the employed camera. First of all, it depends on the camera position and orientation in space. Besides, the projection is affected also by the focal length of the adopted lenses, which is a measure of how much do the lenses converge or diverge the rays they collect [1].

The 2D representation in the image plane is the result of the intersections between the image plane itself and the lines connecting the 3D points of the original object with the optical center of the camera which coincides with the pinhole camera aperture.

The straight line perpendicular to the image plane intersecting the optical center is labeled as the *principal axis* of the camera and the intersection point between the principal axis and the image plane is the *principal point* $\mathbf{p} = (p_x, p_y)$. The focal lenght *f*, i.e. the distance from the lens at which the observed object is perfectly visible and undistorted, corresponds to the distance between the optical center and the principal point in the pinhole camera model.

The model described so far, i.e. the pinhole camera model, makes it possible to map every point in the 3D real space $\mathbf{X} = [X, Y, Z]^T$ to the point in the image plane associated to coordinates $[\frac{fX}{Z}, \frac{fY}{Z}, f]^T$, as it can be seen in figure 2.1.

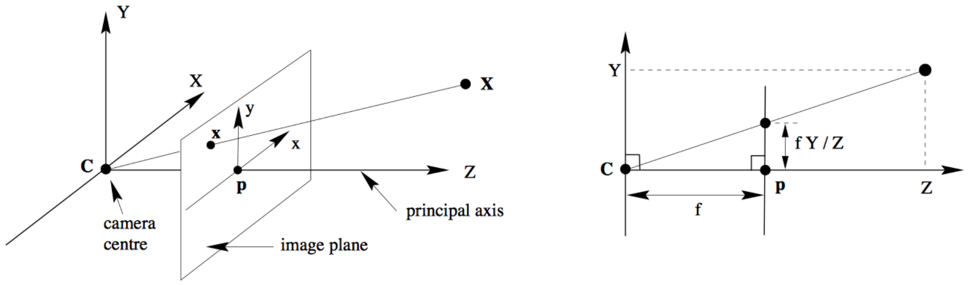


Figure 2.1: Pinhole camera model

This kind of projection is labeled as a *central projection.* Central projections are the projections from a plane to another where the original point and the projected one lie on the same straight line originated from a point which does not belong to either plane. The above mapping from a 3D point to its 2D image is suitable to be expressed in homogeneous coordinates which increase the dimensions of the points by one but also allow a compact representation of sequences of transformations. For instance, the above tranformation can be represented in the following way:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.1}$$

An alternative and more compact representation of the previous projection matrix is $\mathbf{P} = diag(f, f, 1)[I|0]$, where I represents the 3x3 identity matrix and 0 represents the fourth all-zeros column of the projection matrix. Given $\mathbf{X} = [X, Y, Z, 1]^T$ and $\mathbf{x} = [fX, fY, Z]^T$, it is now possibe to rewrite 2.1 as:

$$\mathbf{x} = \mathbf{PX} \tag{2.2}$$

The observations made so far rely on the hypothesis that the central point $\mathbf{p} = (p_x, p_y)$ coincides with the origin of the image plane but it's more customary to set the origin of the image plane in one of its edge points. If this is the case, the image plane coordinates for a 3D point $\mathbf{X}$ are given by $[\frac{fX}{Z} + p_x, \frac{fY}{Z} + p_y, f]$. Hence the projection equation in homogeneous coordinates will now be the following:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.3}$$

It is now possible to define:

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.4}$$

Matrix 2.4 is defined as the *calibration matrix* of the camera. Hence, equation 2.3 can in the following more compact form:

$$\mathbf{x} = \mathbf{KX} \tag{2.5}$$

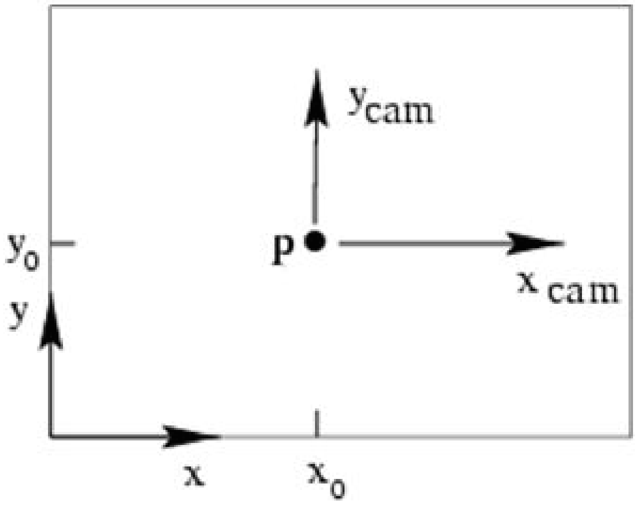


Figure 2.2: Image plane. Origin is set in bottom left corner point

2.1.1 Camera rotations and translations

All the projections seen so far project points from what is defined as the *camera space*, a reference system that is totally integral with the camera, to the image plane. In order to make the camera model more realistic, it is necessary to take into account the possibility that the camera center might be translated with respect to the origin of the reference system and that the camera z-axis coincides with that of the reference system in use. In other words, for the sake of generality, it is preferred to project points from *world space* to the image plan. Therefore, we need to find the mapping between the two spaces.

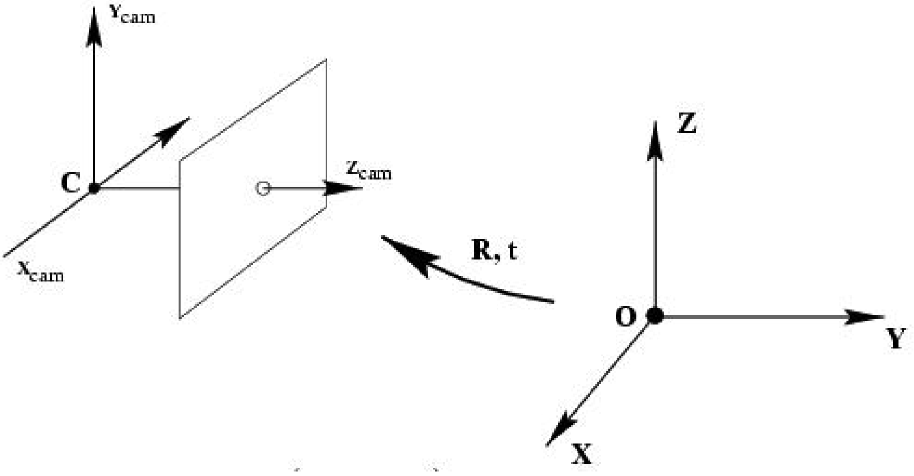


Figure 2.3: Transformation between camera and world space

In general, it is possible to move from one space to the other by means of a rotation and a translation, as figure 2.3 shows. Hence, given $\mathbf{X}$ and $\mathbf{X}_{cam}$, respectively representing the homogeneous coordinates array of a point in the world space and in the camera space, we can write:

$$\mathbf{X_{cam}} = \mathbf{R}(\mathbf{X} - \mathbf{C}) \tag{2.6}$$

where $\mathbf{C}$ is the coordinates array of the camera center and $\mathbf{R}$ is the 3x3 rotation matrix of the camera, both computed with respect to the world space reference system. In homogeneous coordinates and representing the transformation matrix in blocks this becomes:

$$\mathbf{X_{cam}} = \begin{bmatrix} \mathbf{R} & \textbf{-RC} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \textbf{-RC} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X} \tag{2.7}$$

As a consequence of this generalization, the projection matrix which maps points from world space to the image plane is now given by:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \tag{2.8}$$

where $[\mathbf{R}|\mathbf{t}]$ is a 3x4 roto-translational matrix having $\mathbf{t} = [-\mathbf{RC}, 1]^T$ as its fourth column.

Matrix $\mathbf{P}$ has nine degrees of freedom: three of them belong to calibration matrix $\mathbf{K}$, other three correspond to the non-zero values of rotation matrix $\mathbf{R}$ and the last three are the translation coordinates in $\mathbf{C}$. The three values belonging to matrix $\mathbf{K}$ are labeled as *intrinsic parameters*, as they are associated with specific characteristics of the camera in use. The remaining degrees of freedom are the *extrinsic parametes* and they are associated with the camera reference system roto-translation with respect to world space.

2.1.2 CMOS model

The pinhole camera model discussed so far relies on the hypothesis that the different reference systems share the same sizes for their unity, i.e. the scale factor from one to another is always equal to one. However, this is not always the case: it often happens that cameras resize objects with respect to world space dimensions, according to the size of their pixels. For instance, this happens with both old Charge Coupled Device (CCD) and Complementary Metal-Oxide Semiconductors (CMOS) image sensors. These sensors differ not only for their ways of converting light into electrons but also for the scale factors adopted for the images. In particular, CCD sensors may be characterized by not perfectly squared pixels and this is why we employ a CMOS model.

Given these considerations and labeled as m_x and m_y the 2D dimensions of a pixel in this model, camera calibration matrix $\mathbf{K}$ can be rewritten in the following way:

$$\mathbf{K} = \begin{bmatrix} a_x & 0 & x_0 & 0 \\ 0 & a_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.9}$$

where $a_x = m_x f$, $a_y = m_y f$, $x_0 = m_x p_x$ and $y_0 = m_y p_y$.

Since we consider square pixels, $m_x = m_y$; therefore the number of degrees of freedom of the projection matrix $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ is now ten. Besides,for the sake of generality, we introduce a *skew factor* s between axes x and y which is another camera intrinsic parameter that modifies calibration matrix in the following way [2]:

$$\mathbf{K} = \begin{bmatrix} a_x & s & x_0 & 0 \\ 0 & a_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.10}$$

This brings the number of degrees of freedom of the calibration matrix to eleven. Anyways, in many applications, the skew factor is null, suggesting a perfect perpendicularity between the axes in camera space and bringing back the number of degrees of freedom to ten.

2.2 Projections

Projections are an extremely important mathematical tool which play a key role in computer vision and graphics. As we have already outlined before, a projection is the operation through which it is possible to represent a 3D object onto a 2D surface, a screen in most of the cases.

There exists a plethora of different kind of projections and some of them will be now presented in order to prepare the reader to the image analysis operations performed in this project [3] [4].

2.2.1 Central Projections

All the transformations introduced so far belong to the category of *central projections*. These are projections in which the original point and its image lie on parallel planes and on a straight line that starts from a point that does not lie on either plane. The bundle of all the projecting rays has then a point in common, i.e. the center of projection. The projected image lies onto the image plane and it is the result of the intersections between the projecting rays and the plane itself. As it is clear, the camera model discussed before strongly relies on central projections: in particular, with a good approximation, we can treat the camera center as the center of projection, considering that all the rays cross the center of its lens.

In mathematical terms a central projection is a linear transformation from a 3D space to a 2D space, therefore represented by a 3x4 matrix (when using homogeneous coordinates). A generic 3D point in homogeneous coordinates $\mathbf{X} = [X, Y, Z, W]^T$ is mapped onto a 2D point having homogeneous coordinates $\mathbf{x} = [x, y, w]^T$:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \mathbf{P} \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \tag{2.11}$$

From a visual point of view, because of the properties of central projections, all the 3D points sharing the same X, Y and Z values but with different W values are mapped onto the same 2D image point. In fact, all of these points lie on the same straight line that connects the center of projection and the image point itself.

Since central projections flatten the image and reduce its dimension, projection matrix $\mathbf{P}$ is characterized by a 3x3 non-null block and by an all zeros fourth column. Besides, if we set the origin $[0, 0, 0, 1]^T$ as the center of projection, the 3x3 block will become the 3x3 identity matrix.

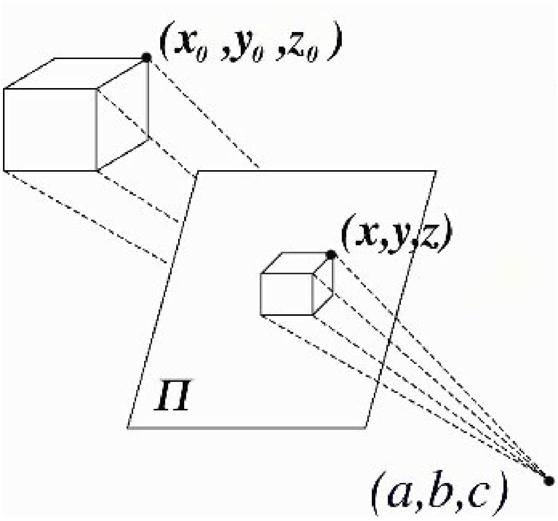


Figure 2.4: Example of a central projection (in Cartesian coordinates)

It may happen that the object to be projected is bidimensional. In this case all of its points belong to a plane, their homogeneous coordinates can be represented with a 3D array (instead of the usual 4D one) and, as a consequence, projection matrix becomes a 3x3 matrix and 2.12 can be rewritten in the following simpler form:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \mathbf{H} \begin{bmatrix} X \\ Y \\ W \end{bmatrix} \tag{2.12}$$

This kind of transformation takes the name of *projective transformation.*

2.2.2 Projective Transformations

A projective planar transformation maps a 3D homogeneous coordinates array onto another one. Thus it can be represented as a 3x3 non-singular matrix:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{2.13}$$

In a more compact form, we can write: **x'** = **Hx**. Matrix **H** is a homogeneous matrix, if it gets multiplied by a non-null scale factor, the transformation performed is not altered; this means that this kind of matrix is defined up to a multiplicative non-null constant.Therefore this matrix has eight degrees of freedom, given by the eight independent ratios between the couples of its nine elements.

A projective transformation preserves collinearity and incidence but it does not preserve parallelism, length and angle. However, projective transformations can be further classified. Some of these transformations, which are of key interest for our project, will be analyzed below.

2.2.2.1 Isometries

Isometries are transformations between metric spaces, i.e. spaces where a distance measurement has been defined. They owe their name to their main and most important feature, which is that of preserving distances.

For instance, an Euclidean plane isometry (adopting the Euclidean distance) maps points from a subspace of $\mathbb{R}^2$ to a different one preserving their relative distances [4].

This kind of transformation can be represented as in the following matrix form:

$$\begin{bmatrix} x_1' \\ x_2' \\ 1 \end{bmatrix} = \begin{bmatrix} \epsilon cos\theta & -sin\theta & t_x \\ \epsilon sin\theta & cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \tag{2.14}$$

where ϵ is a parameter which can be equal to +1 or -1. If $\epsilon = 1$ the isometry preserves orientation and is labeled as a Euclidean isometry. If instead $\epsilon = -1$, it inverts the orientation and is labeled as an Euclidean reflected transformation.

The above transformation has three degrees of freedom, as it can be described with only three parameters: one for the rotation (θ) and two for the translation (t_x and t_y). This kind of isometry is suitable for modeling rigid objects displacements.

2.2.2.2 Similarities

Two geometrical objects are called similar if one can be obtained from the other by uniformly scaling (enlarging or reducing), possibly with additional translation, rotation and reflection. This means that either object can be rescaled, repositioned, and reflected, so as to coincide precisely with the other object. If two objects are similar, each is congruent to the result of a particular uniform scaling of the other.

A similarity (also called a similarity transformation or similitude) of a Euclidean space is a bijection from the space onto itself that multiplies all distances by the same positive real number. This kind of transformation can be represented as in the following matrix form:

$$\begin{bmatrix} x_1' \\ x_2' \\ 1 \end{bmatrix} = \begin{bmatrix} scos\theta & -ssin\theta & t_x \\ ssin\theta & scos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \tag{2.15}$$

or as in the following more compacted one:

$$x' = H_s x = \begin{bmatrix} sR & t \\ t & 1 \end{bmatrix} x \tag{2.16}$$

where s is the isotropic scaling value. Similarities preserve shapes, angles, mutual position of the lines and the ration between lengths but not the absolute lengths. This kind of transformation has four degrees of freedom, one more with respect to isometries due to the presence of the scale factor.

2.2.2.3 Affine transformations

Affine transformations are a subset projective transformations that preserve points, straight lines, and planes. Sets of parallel lines remain parallel after an affine transformation.

Affine transformations are typically used to correct for geometric distortions or deformations that occur with non-ideal camera angles.

They can be represented as in the following matrix form:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a22 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \tag{2.17}$$

or as in the following more compacted one:

$$x' = H_A x = \begin{bmatrix} A & t \\ t & 1 \end{bmatrix} x \tag{2.18}$$

An affine transformation has six degrees of freedom

2.2.3 Direct Linear Transformation (DLT) algorithm

A main issue in computer vision is that of estimating the geometric transformation between two images, given the points in the image plane of the camera. Given the corresponding sets of points x_i and x'_i, the aim is that of estimating the transform to be applied to x_i in order to obtain x'_i. These points belong to two different planes, hence they are represented by two 3D homogeneous coordinates array. As a consequence, the transformation homography is a 3x3 matrix defined up to a multiplicative scale factor. The total number of degrees of freedom is then eight but, since the third element of both points coordinates array is equal to one, we are interested in computing what maps the first two coordinates of the arrays, thus reducing the degrees of freedom of the problem. The image points to be analyzed are usually affected by noise. For this reason several points are used for performing the estimation.

In order to perform the estimation, we implemented the Direct Linear

Transformation (DLT) algorithm. Its main steps are described in the following section.

2.2.3.1 The Direct Linear Transformation algorithm

The matrix equation $x'_i = Hx_i$ can be be rewritten in the cross product form $x'_i \times Hx_i = 0$. If we label the transposed j-th row of the H matrix as h_j^T, we can write:

$$Hx_i = \begin{bmatrix} h_1^T x_i \\ h_2^T x_i \\ h_3^T x_i \end{bmatrix} \tag{2.19}$$

If we define $x'_i = (x'_i, y'_i, w'_i)^T$, the above cross product ca be reformulated as:

$$x'_i \times Hx_i = \begin{bmatrix} y'_i h_3^T x_i - w'_i h_2^T x_i \\ w'_i h_1^T x_i - x'_i h_3^T x_i \\ x'_i h_2 T x_i - y'_i h_1^T x_i \end{bmatrix} \tag{2.20}$$

Knowing that $h_j^T x_i = x^T h_j$ for $j = 1, 2, 3$, we have a set of three equations which be expressed as in the following matrix form:

$$\begin{bmatrix} 0^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \\ -y'_i x_i^T & x'_i x_i^T & 0^T \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0 \tag{2.21}$$

This equation can be re-expressed as $\mathbf{A}_i\mathbf{h} = \mathbf{0}$, where $\mathbf{A}_i$ is a 3x9 matrix and **h** a nine-dimensional elements array which contains the nine elements of matrix **H**. Among the three equations which form the above homogeneous linear system, only two are linearly independent; the third equation can be obtained by a linear combination of the other two. As a consequence, the set of equations to be solved reduces to:

$$\begin{bmatrix} 0^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = 0 \tag{2.22}$$

Now $\mathbf{A}_i$ a 2x9 matrix. Each correspondence between to two points produces an $\mathbf{A}_i$, which get collected as rows in the higher dimensional matrix **A**. Considering four couples of corresponding points, matrix **A** has a rank of eight and a non-banal solution for $textbfh$ can be found.

This solution determines the elements of the homography between the points up to a scale factor, which can be arbitrarily set by choosing $||\mathbf{h}|| = 1$.

With more than four corresponding couples of points the problem becomes over-determined. If the relative positions between corresponding points is exact for all them we can find an exact solution, otherwise an approximated one can be found.

2.3 Camera calibration

Camera calibration process consists in obtaining the main camera parameters in order to remove distortion due to lenses and sensor. These parameters let us trace where a 3D point from world space is projected onto the camera sensor.

2.3.1 Calibration methodology

In order to obtain a three-dimensional reconstruction we need to know how the acquisition system works and the intrinsic and extrinsic characteristics of the camera in use.

2.3.1.1 Intrinsic Parameters

Intrinsic parameters are the features connected with the camera and its technical characteristics, such as focal length, position of the principal point and coefficients of optical distortion. We define: the focal length of the camera lens for both axes (f_x, f_y), the optical center position for the camera sensor (c_x, c_y) (both expressed in pixels) and the distortion coefficients k_1, k_2, p_1, p_2. In an ideal camera, a 3D point (X, Y, Z) would be projected onto coordinates:

$$x = X\frac{f_x}{Z} + c_x \quad , \quad y = Y\frac{f_y}{Z} + c_y \tag{2.23}$$

However camera lenses normally distort the image by enhancing the distance of the points: the more distant the points from the center, the more it gets distorted. Thus, the vertical stripes near the image borders appear to be slightly bent. As a consequence, in order to determine a pixel projection, we must consider the distortion components. Two types of distortions (radial and tangential) are considered and they are represented by the parameters p_1, p_2, k_1, k_2.

2.3.1.2 Extrinsic parameters

Extrinsic parameters are connected with the features that describe the camera with respect to the external reference system. In order to compute the projective transformation for mapping a point onto an arbitrary reference system, it is necessary to know the extrinsic parameters. These include information such as the translation and the rotation of the camera with respect to the origin of the reference system. In other words, the extrinsic parameters are the parameters of the 3D roto-translation that maps the camera reference system onto the arbitrary one. The rotation matrix elements can be obtained from the corresponding parameters by using Rodrigues rotation formula [5].

By using homogeneous coordinates we express both rotation and translation of the camera with a 4x4 invertible matrix whose inverse matrix performs the opposite transform between the same reference systems.

2.3.2 Calibration targets

The calibration of intrinsic parameters is based on the information obtained from the homographies which connect the corresponding sets of coplanar points. It is quite common to adopt a regular pattern target image in order to extract these information. A rectangular pattern of white and black squares is often suitable.

The main advantage of using this kind of target is that it is easily identified: its realization does not require high requirements, it can be produced by any graphic software and printed using a good quality laser printer.

The single squares or rectangles do not have to be necessarily black and white but this coloring solution allows a simpler detection of the corners, essential for the camera calibration. On the other hand it is of key importance that the target pattern is flat and applied onto a flat and smooth surface. During calibration process performed in this project. we printed the pattern on a A4 paper with each square having a side of 23 mm and we attached it to an plastic flat surface.

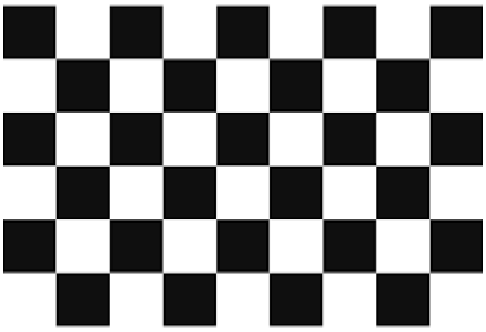


Figure 2.5: Example of a calibration pattern with 8×5 vertices (9×6 squares)

2.3.3 Extraction of the features from the planar pattern

In a planar target like the one described above the information used for the calibration, is given by the coordinates of the intersection points between the corners of the squares of the chessboard pattern [6]. The target has to be completely visible from the camera. The first step of the extraction process of the coordinates is the conversion of the image in binary form, which includes converting it in grey scale using an algorithm based on adaptive threshold.

Defined with $I_{r,g,b}(x, y)$ the value of the red, green and blue component of each pixel in the original image and $I_y(x, y)$ the value of luminance (from 0 to 255), we have that:

$$I_y(x, y) = \alpha I_r(x, y) + \beta I_g(x, y) + \gamma I_b(x, y) \tag{2.24}$$

where (α, β, γ) are the coefficients needed to calculate the luminance exploiting the radiometric model of the human eye and they respectively have the value of (0.299, 0.587, 0.114).

Once the grey scale image is obtained, the binarization thresholds can be computed. Given the mean luminance value of the image μ_1 as

$$\mu_1 = \sum_{x=0}^{M} sum_{y=0}^{N} \frac{I_y(x, y)}{NM} \tag{2.25}$$

where M and N are the dimensions of the image along x and y directions, expressed in pixels, the threshold value B is computed as:

$$B = \begin{cases} \mu_1, & \text{if } \mu_1 - Th > Th \\ 10, & \text{if } \mu_1 - Th <= Th \end{cases} \tag{2.26}$$

where Th is a threshold sensibility parameter, estimated to be approximately equal to ten. A *Canny edge detector* [7] is then applied to the image according to the computed threshold: this is a technique, based on an iterative algorithm, that allows extracting useful structural information from the image and dramatically reducing the amount of data to be processed [8]. When all the edges have been detected, they can be analyzed in order to detect those ones that form convex quadrangular polygons. In order to avoid errors, the algorithm automatically removes all the polygons having a perimeter smaller than a given threshold whose value depends on the size of the target image.

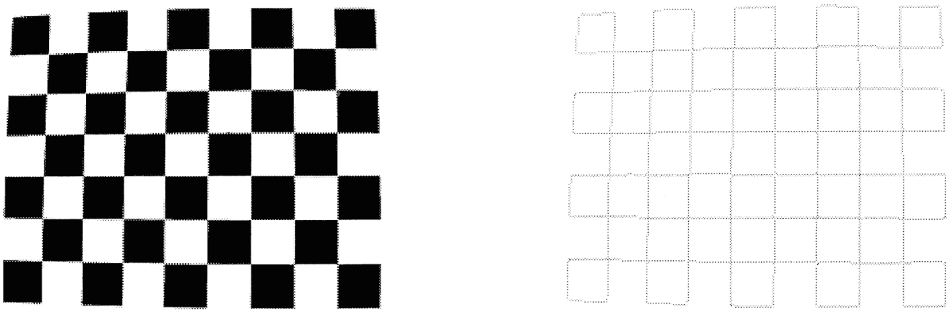


Figure 2.6: Edge detection result

Every time the algorithm detects an edge, it stores the coordinates of its end points. These coordinates can then be sorted in order to uniquely associate each corner point to a specific position in the chessboard, following a matrix like scheme similar to the one below.

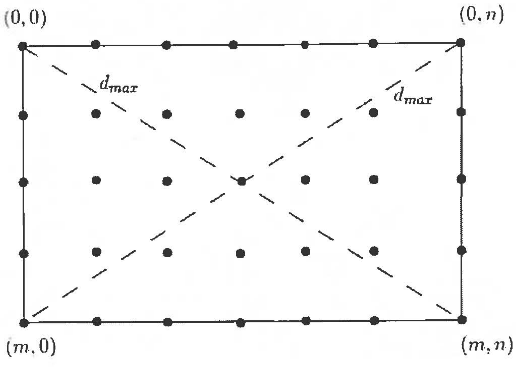


Figure 2.7: Detected points matrix for $m \times n$ chessboard pattern

The sorting algorithm is an iterative procedure: the first step is that of finding the four points which have the longest distances between each other, corresponding to the corner points closest to the edges of the chessboard. The second step is that of determining the points belonging to the chessboard perimeter by exploiting the prior knowledge of the sizes of the target in terms of number of squares per dimension. The algorithm then determines all the edge points coordinates by considering all the internal perimeters, obtained by iteratively reducing both dimensions by one, until the last and most central edge point is detected. The coordinates estimated with this procedure are expressed with a pixel precision but there exist other algorithms which are able to perform this estimation even with a sub-pixel precision. For instance, *Harris corners detector* can be applied to the already processed image in order to increase the precision of the coordinates, applying it onto $w \times w$ windows centered on the point previously identified as the corner point.

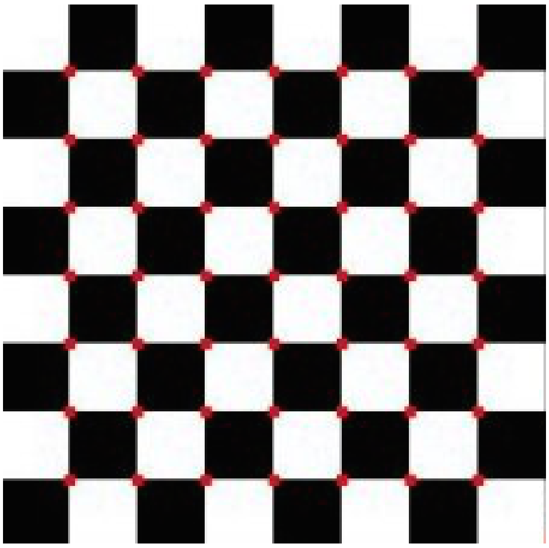


Figure 2.8: Chessboard target processed with Harris detector

2.3.4 Camera model

Once the planar target has been processed and the necessary information has been extracted, it is possible to proceed to estimate the intrinsic parameters of the cameras in use. The basic model adopted is the CMOS model as described above. Now given a generic point in world space $\mathbf{X} = (x, y, z, 1)$, its projection onto the camera plane is given by the following transformation:

$$\beta \mathbf{x} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \mathbf{X} \tag{2.27}$$

where $\mathbf{K}$ is the intrinsic parameters matrix, β is an arbitrary scale factor and $\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$ is the extrinsic parameters matrix.

2.3.5 Homography derived from target

A generic point X belonging to a three-dimensional planar object can be expressed as $(X, Y, k, 1)$ with k constant. Assuming $k = 0$, the previous equation can be updated as:

$$\beta \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} r_1 & r_2 & r_3 & t] \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.28}$$

where r_1, r_2, r_3 are the column arrays of the rotation matrix $\mathbf{R}$. Points $\mathbf{X}$ and $\mathbf{x}$ are linked by an homography $\mathbf{H}$:

$$\beta \mathbf{x} = \mathbf{H}\mathbf{X} \tag{2.29}$$

where $\mathbf{H} = \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$.

Given the camera image of a planar target, it is possible to estimate the homography $\mathbf{H}$ that can project $\mathbf{X}$ onto $\mathbf{x}$. Matrix $\mathbf{H}$ is estimated by determining the homography that minimizes the euclidean distance between the computed point $\mathbf{x}$ and the estimated one, labeled as $\tilde{\mathbf{x}}$. Hence the estimated homography is given by:

$$\hat{\mathbf{H}} = argmin_H \{ \sum d(\mathbf{x}_i, \mathbf{H}\mathbf{X}_i)^2 \} \tag{2.30}$$

Recalling that $\mathbf{H} = \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$ and that r_1 and r_2 are orthonormal, we can extrapolate the following equations:

$$h_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} h_2 = 0 \tag{2.31}$$

$$h_1^T \mathbf{K}^{-T} \mathbf{K} h_1 = h_2^T \mathbf{K}^{-T} \mathbf{K}_2^h \tag{2.32}$$

The model of the plane used to estimate $\mathbf{H}$ can be expressed by the following equation:

$$\begin{bmatrix} r_3 & e_3^T t \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = 0 \tag{2.33}$$

where $w = 0$ if we consider points at infinite distance and $w = 1$ for all other cases. This plane intercepts the infinite plane in a line. $(\mathbf{r}_1, 0)^T$ and $(\mathbf{r}_2, 0)^T$ belong to this line. Each point of such a line can then be expressed as a linear combination of these two points:

$$X_\infty = a \begin{bmatrix} r_1 \\ 0 \end{bmatrix} + b \begin{bmatrix} r_2 \\ 0 \end{bmatrix} = \begin{bmatrix} ar_1 + br_2 \\ 0 \end{bmatrix} \tag{2.34}$$

X_∞ point is labeled as the circular points, satisfying the equation $X_\infty^T X_\infty = 0$, implying that:

$$(ar_1 + br_2)^T (ar_1 + br_2) \vee a^2 + b^2 = 0 \tag{2.35}$$

A solution to last equation is $b = \pm ia$. This leads to:

$$X_\infty = a \begin{bmatrix} r_1 \pm ir_2 \\ 0 \end{bmatrix} \tag{2.36}$$

The projections of these points onto the image plane, ignoring the β scale factor, is then given by:

$$x_\infty = \mathbf{K}(r_1 \pm ir_2) = h_1 \pm ih_2 \tag{2.37}$$

The point x_∞ belongs to the absolute conic, described by the equation $\omega = \mathbf{K}^{-T}\mathbf{K}^{-1}$. This leads back to the solving equations given above for extracting h_1 and h_2 values.

Considering that the homography has eight degrees of freedom, given the six degrees of freedom of the extrinsic parameters (three for rotation and three for translation) we deduce that each homography contributes to define just two intrinsic parameters.

2.3.6 Extrinsic parameters estimation

Matrix $\mathbf{K}_{-T}\mathbf{K}_{-1}$ describes the absolute conic of the camera, defined as the following 3x3 matrix:

$$\mathbf{B} = \mathbf{K}_{-T}\mathbf{K}_{-1} = \begin{bmatrix} \frac{1}{f^2} & -\frac{s}{f^2} & -\frac{y_0 s - x_0 f}{f^3} \\ -\frac{s}{f^3} & \frac{s^2}{f^4} - \frac{1}{f^2} & -\frac{s(y_0 s - x_0 f)}{f^2} - \frac{y_0}{f^2} \\ \frac{y_0 s - x_0 f}{f^3} & -\frac{s(y_0 s - x_0 f)}{f^2} - \frac{y_0}{f^2} & \frac{(y_0 s - x_0 f)^2}{f^4} + \frac{y_0^2}{f^2} + 1 \end{bmatrix} \tag{2.38}$$

where f is the focal length of the camera lenses system and s is the skew factor. Matrix $\mathbf{B}$ is symmetric and can be more synthetically re-express as the following array:

$$b = \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix} \tag{2.39}$$

Given this, equation 2.3.5 can now be written as:

$$h_i^T \mathbf{B} h_j = \mathbf{v}_{ij}^T b \tag{2.40}$$

where $\mathbf{v}_{ij}$ corresponds to the following array:

$$b = \begin{bmatrix} h_{i1}h_{j1} \\ h_{i1}h_{j2} + h_{i2}h_{j1} \\ h_{i2}h_{j2} \\ h_{i1}h_{j3} + h_{i3}h_{j1} \\ h_{i2}h_{j3} + h_{i3}h_{j2} \\ h_{i3}h_{j3} \end{bmatrix} \tag{2.41}$$

If we substitute $h_1^T \mathbf{K}^{-T}\mathbf{K}^{-1} h_2 = 0$ in this array, we end up obtaining two homogeneous equations:

$$\begin{bmatrix} V_{ij}^T \\ (v_{11} - v_{22})^T \end{bmatrix} = 0 \tag{2.42}$$

If we then combine n of these, we obtain a system having the form:

$$\mathbf{V}b = 0 \tag{2.43}$$

where V is a matrix of dimension $2n \times 6$. As mentioned before, the internal matrix calibration K has 5 degrees of freedom, reduced to 4 in we consider the model adopted, with the same focal length in the two directions.Each view lets us to recover just 2 parameters and so we need at least two views to recover completely the matrix K, ignoring a scale factor. We can add also another equation to the $2n \times 6$ to set the skew value to zero. The solution of the system corresponds to the eigenvector with the smallest associated eigenvalue. The b array is determined as the null right space of $\mathbf{V}^T$, given the singular value decomposition $SVD(\mathbf{V}\mathbf{V}^T = \mathbf{U}\mathbf{W}\mathbf{V}^T$. Once b has been determined, the elements of K matrix can be computed, according to the following equations:

$$y_0 = \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \tag{2.44}$$

$$\alpha = B_{33} - \frac{B_{13}^2 + y_0(B_{12}B_{13} - B_{11}B_{23})}{B_{11}} \tag{2.45}$$

$$f = \sqrt{\frac{\alpha B_{11}}{B_{11} - B_{22} - B_{12}^2}} \tag{2.46}$$

$$s = -\frac{B_{12}f^3}{\alpha} \tag{2.47}$$

$$x_0 = \frac{sy_0}{f} - \frac{B_{13}f^2}{\alpha} \tag{2.48}$$

It's now possible to determine the camera extrinsic parameters, which describe camera roto-translation with respect to the origin of the world reference system.

Recalling that $\mathbf{H} = \mathbf{K}\begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$ and that $h_1^t\mathbf{K}^{-T}\mathbf{K}^{-1}h_2 = 0$, we have that:

$$r_1 = \gamma\mathbf{K}^{-1}h_1 \tag{2.49}$$

$$r_2 = \gamma\mathbf{K}^{-1}h_2 \tag{2.50}$$

$$r_3 = r_1 \times r_2 \tag{2.51}$$

$$t = \gamma \mathbf{K}^{-1} h_3 \tag{2.52}$$

where γ is a constant scale factor. Also in this case, as it happened in the extrapolation for the intrinsic parameters, we need a cost function to minimize in order to find the parameters that best suit the correspondence between points in world space and points in camera space.

As we mentioned before, a point belongs to the planar target if its estimated coordinates satisfy the equation:

$$\hat{\mathbf{x}} = \mathbf{K} \left[\begin{array}{c} \mathbf{Rt} \end{array} \right] \mathbf{X} \tag{2.53}$$

We then define the re-projection error as the difference between the computed coordinates (using the found matrix values) and the detected ones:

$$e = \mathbf{x} - \hat{\mathbf{x}} \tag{2.54}$$

Since we are dealing with bidimensional points, Euclidean distance can be used as error measurement for the points estimation. The re-projection error can then be defined as the norm of the difference between the same homogeneous coordinates arrays:

$$e = ||\mathbf{x} - \hat{\mathbf{x}}|| \tag{2.55}$$

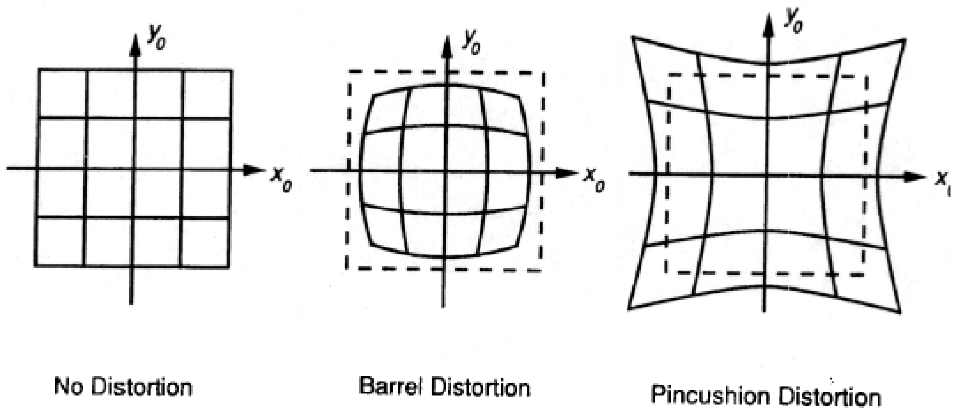
This error is suitable to be used as cost function for determining the parameters of the camera, according to the following formula:

$$argmin_{\mathbf{k},\mathbf{R},\mathbf{t}} \Big\{ \sum_{i=1}^{N} \sum_{j=1}^{L} ||x_{ij} - \mathbf{K} \left[\begin{array}{cc} \mathbf{R}_i & \mathbf{t}_i \end{array} \right] \mathbf{X}_i||^2 \Big\} \tag{2.56}$$

The solution to this problem can be obtained by minimizing the given cost function and applying Nelder-Mead simplex method [9]. In order to reduce the number of parameters, rotation matrix $\mathbf{R}_i$ can be expressed as an array of rotation angles for the reference system axes $\mathbf{r}_i$. It is possible to switch from $\mathbf{R}_i$ to $\mathbf{r}_i$ by using Rodrigues rotation formula.

2.3.7 Radial distortion

Representing the points according to the camera projection equation considered so far $\beta\hat{\mathbf{x}} = \mathbf{K}\begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix}\hat{\mathbf{X}}$ is not yet the most accurate representation of the points measured during the acquisition phase. In fact, an element that has not been taken into account is the distortion contribution introduced by the adopted cameras. Among the different kinds of optical distortion, the most relevant one in this kind of application is the radial one. Radial distortion can mainly occur in two different ways: "*barrel*" distortion and "*pincushion*" distortion, schematized in the following figure:



No Distortion

Barrel Distortion

Pincushion Distortion

Figure 2.9: Effects of the different kinds of radial distortion

In barrel distortion, image magnification decreases with distance from the optical axis. The apparent effect is that of an image which has been mapped around a sphere (or barrel). Fisheye lenses, which take hemispherical views, utilize this type of distortion as a way to map an infinitely wide object plane into a finite image area. In a zoom lens barrel distortion appears in the middle of the lens's focal length range and is worst at the wide-angle end of the range. In pincushion distortion, image magnification increases with the distance from the optical axis. The visible effect is that lines that do not go through the center of the image are bowed inwards, towards the center of the image, like a pincushion. A mixture of the two types of radial distortion, referred to as "*mustache*" distortion, may sometimes occur. In this case the resulting image starts out as barrel distortion close to the image center and gradually turns into pincushion distortion towards the image periphery, making horizontal lines in the top half of the frame look like a handlebar mustache [10].

Mathematically, barrel and pincushion distortion are quadratic, meaning they increase as the square of distance from the center. In mustache distortion the quartic (degree 4) term is significant: in the center, the degree 2 barrel distortion is dominant, while at the edge the degree 4 distortion in the pincushion direction dominates. Other distortions are in principle possible – pincushion in center and barrel at the edge, or higher order distortions (degree 6, degree 8) – but do not generally occur in practical lenses, and higher order distortions are small relative to the main barrel and pincushion effects.

In order to be able to remove or at least attenuate the effects of radial distortion, it is first of all necessary to determine the parametric mathematical model that best describes the actual distortion performed by the adopted camera and estimate the corresponding parameters.

The projected point affected by distortion in the camera plane $\mathbf{x}_d = (u_d, v_d, 1)$ is related to the ideal undistorted point $\mathbf{x} = (u, v, 1)$ by the the following relationship:

$$r_d = r + \delta_r \tag{2.57}$$

where r_d is the radial distance of the distorted point, obtained as the sum of the undistorted radial distance r and the radial distortion contribution δ_r. Photogrammetry studies have defined the standard of the mathematical structure of the function of radial distortion which can be described by a polynomial expression:

$$r_d = rf(r) = r(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + ...) \tag{2.58}$$

with k_i being the distortion coefficients. Given $\mathbf{x}_d = (x_d, y_d, 1)$ defined as:

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \mathbf{K}^{-1} \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} \tag{2.59}$$

the radial distance of the distorted point $\mathbf{x}$ can now be expressed as :

$$r_d^2 = x_d^2 + y_d^2 \tag{2.60}$$

The point distorted coordinates then can be expressed as:

$$x_d = xf(r) \tag{2.61}$$

$$y_d = yf(r) \tag{2.62}$$

As we mentioned before, the strongest contribution in the polynomial model of radial distortion comes from the quadratic term. Since adopting a too complex polynomial model may lead to instabilities in the estimation phase, we have decided to consider only the terms up to the fourth order. Hence, the adopted model is given by:

$$f(r) = 1 + k_1 r^2 + k_2 r^4 \tag{2.63}$$

Applying this distortion model to a generic unditorted point $\mathbf{x} = (u, v, 1)$, we get the following distorted coordinates:

$$u_d = (u - x_0)f(r) + x_0 \tag{2.64}$$

$$v_d = (v - y_0)f(r) + y_0 \tag{2.65}$$

where radial distortion has been applied, after setting the principal point of the camera (x_0, y_0) as the center of distortion.

The estimation process for the distortion coefficients k_1 and k_2 is similar to that performed for the other parameters computed so far. The coefficients are then those which minimize the following cost function:

$$argmin_{k_1,k_2}\Big\{\sum_{i=1}^{N}\sum_{j=1}^{L}||x_{ij} - x_{dij}(k_1, k_2)||^2\Big\} \tag{2.66}$$

2.3.8 Linear Optimum Estimator

The two estimators computed so far, one for the camera intrinsic and extrinsic parameters (2.56) and one for the radial distortion parameters (2.66) can be combined in order to obtain a full parameters estimation for the adopted camera. Alternatively, it is possible the estimate the aforementioned parameters with an iterative algorithm which alternates steps of estimation of the two groups of parameters until convergence.

Since these approaches have proved to be equivalent in terms of quality of the found solution, we have decided to adopt the most compact form of the problem which also has the smallest convergence times.

$$argmin_{\mathbf{K},\mathbf{R}_i,\mathbf{t}_i,k_1,k_2}\Big\{\sum_{i=1}^{N}\sum_{j=1}^{L}||x_{ij}-x_{dij}(\mathbf{K},\mathbf{R}_i,\mathbf{t}_i,k_1,k_2)||^2\Big\} \tag{2.67}$$

The minimization of this cost function can be classified as a problem of bundle adjustment [11]. Exploiting the geometric properties of the target we can reduce the dimension of the problem of a factor 6/L. This is possible because a group of L points of the target can be represented with just 6 parameters 3 of rotation and 3 of translation. A complete bundle adjustment on N views of L points would involve a not linear estimation of 4 intrinsic parameters, 2 coefficients of distortion and $N \times L$ parameters for all the three-dimensional reconstructed points.

The estimator used in this study involves the estimation of:

- four intrinsic camera parameters ($\mathbf{K}$);
- two distortion coefficients (k_1, k_2);
- six extrinsic parameters for the rotation matrix ($\mathbf{R}$);
- three parameters for the translation array ($\mathbf{t}$);

Minimization is performed applying Nelder-Mead simplex method, initializing the parameters to some initial values obtained through a preliminary closed-form computation [12]. The initial values of both the distortion coefficients is set to zero. These settings and this procedure have proved to provide a good estimation of the camera model.

2.3.9 Calibration using OpenCV

OpenCV (*Open Source Computer Vision Library*) is an open source computer vision and machine learning software library. It was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

Since an efficient camera calibration is a basic requirement for many applications, there exists a lot of calibrations tools already available and OpenCV provides its own algorithm. The OpenCV implementation is basically the C++ adaption of the *Camera Calibration Toolbox* available in Matlab.

In order to estimate all the camera parameters, a calibration pattern is needed. OpenCV calibration process adopts a flat checkered rectangular pattern, very similar to the chessboard pattern considered so far. Besides, it necessary to be aware of the sizes of the pattern blocks, which need to be measured in advance and and supplied to the calibration algorithm. Once these preliminary settings have been set, the next step is that of capturing several images of the calibration pattern with the camera. The pattern should be showed to the camera system at different locations and orientations (provided it is fully visible to the camera), in order to obtain a more robust and reliable calibration.

In general, the more images are taken for camera calibration, the better is the calibration. However, fifteen to twenty images usually provide good results. Once a sufficiently large sequence of suitable for calibration images is captured, they can be processed via OpenCV or Matlab toolbox.

In order to perform an effective calibration, it is necessary to use different pattern perspectives and orientations: at least onefrontal one and four where the border of the pattern is near the image border. Placing the pattern near the image border helps a more accurates estimation of camera distortion coefficients.

The main advantage of the OpenCV implementation is that it provides automatic detection of all chessboard-corners. Especially with large calibration image sequences this becomes quite convenient. Besides, the adopted algorithm simultaneously estimates intrinsic and extrinsic parameters of the camera.

As output, the program generates a .yml file that can be read and processed with *ArUco*, an open source library for camera pose estimation, written in C++ and based on the use of squared markers [13] [14].

2.4 Markers

A marker system consists of a set of patterns that can be detected by a computer equipped with a camera and an appropriate detection algorithm. Markers placed in the environment provide easily detectable visual cues for many applications, including indoor tracking, robot navigation, augmented reality and, in general, all applications where the relative pose between a camera and an object is required [15].

Markers can be classified according to different features. One main distinction is that between active and passive markers and their relative detection system. Active markers are infrared light emitting elements, mostly LEDs. Each LED is connected to a led control unit transported by the object to be tracked and activated stroboscopically by a cable or telemetric multiplexed signal. The extraction of the markers in active systems is quite straightforward since there's high contrast between the activated markers and the background. Besides, tracking is also made simple because only one LED at a time is activated. A drawback of active markers systems is that is their applicability is reduced by the mechanical restraints imposed by the LED-wiring attached to the subject of the calibration and tracking process

Passive markers most commonly used in tracking systems, are retro-reflectors. These markers reflect incoming infrared radiation back in the direction of the incoming light. More precisely the IR radiation is back-reflected into a narrow range of angles around the source of the incoming light. These markers are mostly spheres covered with retro-reflecting foils, but can also be stickers made from retro-reflecting material.

Passive infrared systems operate by recording the detection of reflected light. Despite also in this case the images obtained by cameras display a pronounced contrast between the markers and the background, the simultaneous presence of many markers in the images requires the use of labeling algorithms that allow a unique identification of the markers along the image sequences. The proximity between two different markers in the image may sometimes complicate this process.

Marker systems can be classified not only by their different working systems but in many other ways. A particular distinction that is important to make is now that between *fiducial* and *not fiducial* markers.

2.4.1 Not-fiducial marker systems

Not-fiducial marker systems consist of markers which carry information about the object they are attached to but they are not meant to provide localization information. Bar codes are an example of not fiducial marker systems. That of encoding information onto a planar surface suitable for detection with optical techniques is a very common methodology. Two-dimensional bar codes can be read by using a passive camera instead of an active laser.

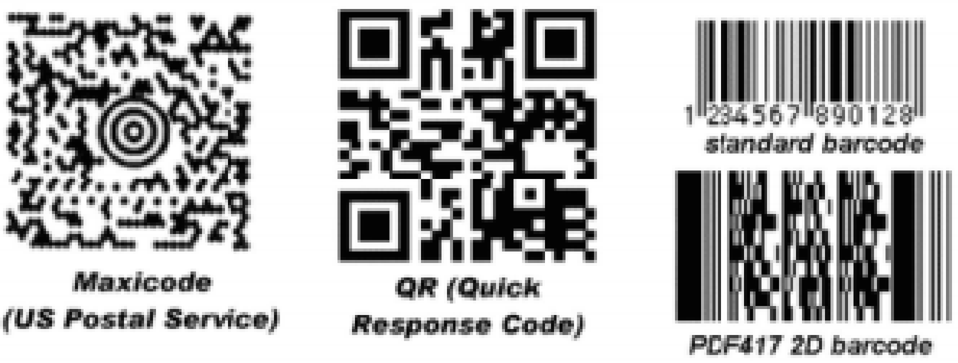


Figure 2.10: Examples of 1D and 2D not fiducial marker systems

This kind of markers does not perform efficiently, as it is not meant for it, in large view fields configuration. In particular, the images would be affected by perspective distortion and, when detected, they do not provide enough image points for three-dimensional pose estimation.

2.4.2 Fiducial marker systems

Fiducial marker systems consist of different unique patterns that are mounted in the environment and automatically detected in digital camera images using an accompanying detection algorithm. They are useful for augmented reality (AR), robot navigation and general applications where the relative pose between a camera and object is required. Important parameters for such marker systems is their false detection rate (false positive rate), their inter-marker confusion rate, minimal detection size (in pixels) and immunity to lighting variation. The various patterns should be distinct enough in order not to be confused with the environment. Ideally, the system should have a library of many unique markers that can be distinguished one from another and the image processing should be robust enough to find the markers in situations of uncontrolled lighting, image noise and blurring, unknown scale, and partial occlusion [16]. Some examples of fiducial markers are given in figure 2.11.

One of the simplest possible approaches with fiducial markers is that of using "point-like" markers, such as LEDs, retro-reflective spheres or planar dots which can be segmented using basic techniques over controlled conditions. Another approach is that of using planar circular markers which encode identification in circular sectors or concentric rings. However, circular markers usually provide just one correspondence point

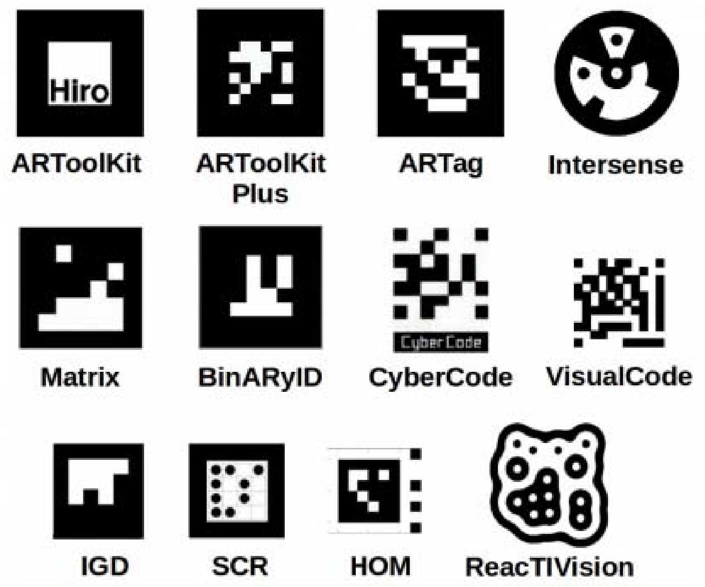


Figure 2.11: Examples of fiducial marker systems

(the center), requiring the detection of several of them for correct pose estimation.

Another valid fiducial markers configuration is that of square-based fiducial marker systems. The main advantage of this approach is that the presence of four prominent points can be employed to obtain the pose, while the inner region is used for identification (either using a binary code or an arbitrary pattern such as a simple image). In the arbitrary pattern category, one of the most popular systems is *ARToolKit*, an open source project which has been extensively used in the last decade, especially in the academic community [17]. ARToolkit markers are characterized by a wide black border with an inner image which is stored in a database of valid patterns. However, despite its popularity, it has some drawbacks. First, it uses a template matching approach to identify markers, obtaining high false positive and inter-marker confusion rates. Second, this system uses a fixed global threshold to detect squares, making it very sensitive to varying lighting conditions. Most of the square-based fiducial systems uses binary codes.

The *ARTag* system is based on the usage of a binary code with redundant bits for error detection. The robustness to lighting and partial occlusion is improved with respect to ARToolkit thanks to an edge-based square detection method, outperforming the fixed threshold one. ARTag uses a binary coding scheme that includes checksum bits for error detection and correction. It also recommends using its dictionary markers in a specific order so as to maximize the inter-marker distances. The main drawback of this approach is that the proposed marker dictionary is fixed to 36 bits and the maximum number of erroneous bits that can be corrected is two, independently of the inter-marker distances of the subset of markers used.

The passive markers adopted in our study are the AR markers: binary coded square-based fiducial markers that are easy to detect and allow achieving a fast and precise detection. An AR Marker, like the one used in ArUco, is a synthetic square marker composed by a wide black border and a inner binary matrix (which determines its identifier) that can be easily detected. The black border facilitates a fast detection in the image and the binary codification allows its identification and the application of error detection and correction techniques. The marker size determines the size of the internal matrix.

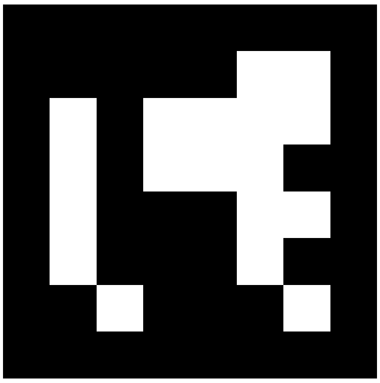


Figure 2.12: An example of marker adopted in the ArUco library

One of great advantages of this marker system is that, instead of using a predefined set of markers, it has a method for generating configurable marker dictionaries (with arbitrary size and number of markers), containing only the number of markers required. The algorithm produces markers using a criterion that maximizes the inter-marker distances and the number of bit transitions.

Markers will be used to calculate the extrinsic parameters of the camera so that it will be possible to 3D render the image, given the origin of the world reference system is located. Since the AR marker is very easy to detect, real time processing and rendering are achievable but it was not necessary in our application, hence we preferred storing all the frame and post-processing them. It must be noted that a marker can be found rotated in the environment but the detection process is able to determine its original rotation, so that each corner is identified unequivocally. Another reason behind the choice of such an approach is that it provides an efficient method for detecting and correcting errors, based on the dictionary obtained. This method allows error correction of a greater number of erroneous bits compared to the other method previously introduced.

2.4.3 Marker coding

As it was mentioned previously, among the most relevant aspects to consider when designing a markers dictionary there are the false positive and negative rates, the inter-marker confusion rate and the number of valid markers. The first two are often tackled in literature using error detection and correction bits, which, on the other hand, reduce the number of valid markers. For what concerns the confusion rate, it mainly depends on the distance between the markers employed. If they are too close, a few erroneous bits can lead to another valid marker of the dictionary, and the error could not be even detected. A desirable feature for markers dictionary is that of having a high number of bit transitions between the different markers. This makes the markers less likely to be confused with elements belonging to the environment. In fact, binary codes with only zeros or ones will be respectively printed as completely black or white markers which can be easily confused with environment objects.

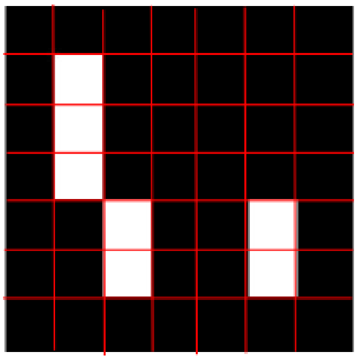


Figure 2.13: An example of marker coding

In figure 2.13 a square-based marker is displayed. Its internal code is given by five words composed by five bits each. In total, each word has only two bits of information out of the five bits available. The other three are employed for error detection. As a consequence, we can have up to 1024 different ids.

The employed coding scheme is a slight variation of Hamming code. The main difference between them is that the first bit (encoding parity of bits three and five) is inverted. The reason behind this choice is that of preventing a completely black rectangle from being a valid marker id. This helps reducing the likelihood of false positive detection from environment elements.

As it was mentioned before, ArUco library allows the possibility to build customized markers dictionaries but in our application this was not necessary as the pre-computed dictionary "DICT-4x4-50" perfectly fit the task. This dictionary includes fifty markers, each being encoded in sixteen bits. The reasons for the choice of this dictionary are different: first of all, fifty markers are more than enough for our study; second, with this limited number of markers it is possible to make the detection process quite fast; lastly, this choice also implies a large inter-marker distance which is desirable in order to increase error rejections in noisy acquisitions.

Another advantage of this approach with respect to typical Motion Capture (MoCap) systems consist in the absence of powered and/or heavy and cumbersome markers like wearable cameras or accelerometers. Furthermore, every single marker provides much more information with respect to approaches based on simple reflective markers since for every marker we are able to accurately estimate its distance from the camera and its spatial orientation.

3
Postural Assessment

Worker health is a serious issue in all job categories. Musculoskeletal disorders, in particular, constitute a major category of worker injury for many of these categories. The repetitive movements, awkward postures, and forceful exertions often involved in a work are the leading causes of this type of injury. To reduce the number of these injuries, worker activities must be tracked and analyzed. Traditional methods to measure work activities rely upon manual on-site observations which are time-consuming and inefficient. To address these limitations, computer vision techniques for worker motion analysis are proposed to automatically identify non-ergonomic postures and movements without on-site work interruption [18].

The process of estimation of human body posture is an essential instrument for various applications, including medical analyses, domotics, animation and many more. The computer vision methods for 3D body posture estimation can be divided in two main categories: marker-based and and marker-free methods. In a marker-based approach, markers are attached to the monitored human body giving particular attention to the most significant body point. These points can be different from one application to another but they often include the joints and the head, as they are some of the most movable parts of the body.

On the other hand, in a marker-free approach, human posture is monitored and analyzed without the aid of markers. Even if they result less accurate when compared to marker-based approaches, these methods reveal to be simpler and still quite effective when it comes to locating significant body points in 3D.

There are several setups that can be used for human posture detection: the one we decided to adopt provides for the usage of two cameras without the aid of a depth sensor. Another effective setup contemplates the usage of a Microsoft *Kinect* device: this method relies only on one and an infrared-based depth sensor.

3.1 MSDs risk factors

The problem of recognizing and classifying impostures working postures comes across as quite complex. In fact, well-controlled epidemiological studies are few and present knowledge about is mainly based on case reports and common sense.

In literature, before the early 2000s, improper postures were almost exclusively associated with physically tiring jobs - such as construction [18] - in which disorders of the musculoskeletal system are often overrepresented. Poor postures and the physical demands of the job were regarded as the causes of the injuries. Only later, the relationship between ergonomic deficiencies in the workplace and diseases of the musculoskeletal system has been discussed [19] and illustrative case reports have been published [20]. Finally, nowadays it is well known that the risk of musculoskeletal injuries and disorders is also very high for people employed in occupations involving repetitive movements and uncomfortable positions [21].

Dentistry is one of the sectors affected by this high musculoskeletal disorders risk [22]. These disorders can result in pain and dysfunction of neck, back, hands and fingers.

The most common factors in work-related musculoskeletal injuries are:

- Repetitive movements;
- Awkward positions;

- Permanence in the same position for extended periods of time;
- Poor postural muscle strength;
- Poor flexibility;
- Stress;
- Infrequent breaks;

The risk for fingers and hand injuries is particularly high only for some specific procedures which require repetitive movements together with pressure application. Instead, due to the necessity of keeping the same position for significant lengths of time, poor posture is a high risk factor in all procedure. For this reason, we decided to investigate on head, neck, shoulders and back position during the operations, with the purpose of tracking all of them and detecting the improper ones.

3.2 Body landmarks

Among the most critical decisions to be made regarding our working approach there were those regarding the markers positions on the monitored body,the solution to be adopted for attaching the markers and, above all, the number of markers to use in order to obtain reliable data for the 3D reconstruction. For what concerns the positions of the markers, it was decided to cover all the body landmarks that could be useful for our model, therefore focusing only on the upper part of the body, from head to sacrum. In particular, we decided to place two markers on the back of the surgical cap in order to track the position of the head.

In order to track the upper body, one marker was placed at the bottom of the neck, two between the shoulder blades following along the spinal column and three in the lumbar regions also along the spinal column. Besides, two markers were placed on the back in correspondence of the clavicles and two more in correspondence of the shoulders. Finally, two lines of three markers each were set on the back, parallel to the spinal column marker line, on both sides. The final markers scheme used for the operations monitoring is shown in figure 3.2.

As it is clear from figure 3.2, the markers are attached on the back of a white elastic t-shirt to be worn by the dentist during all operations. Each

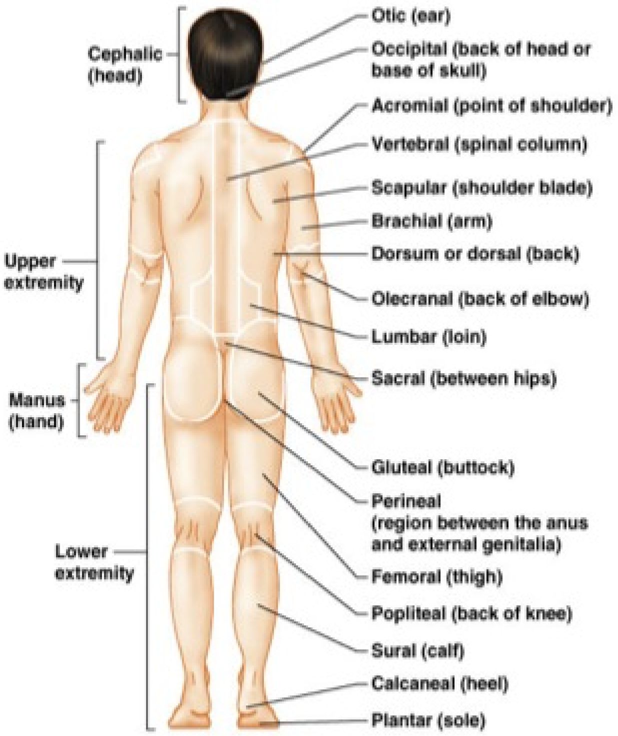


Figure 3.1: Posterior view of body regions

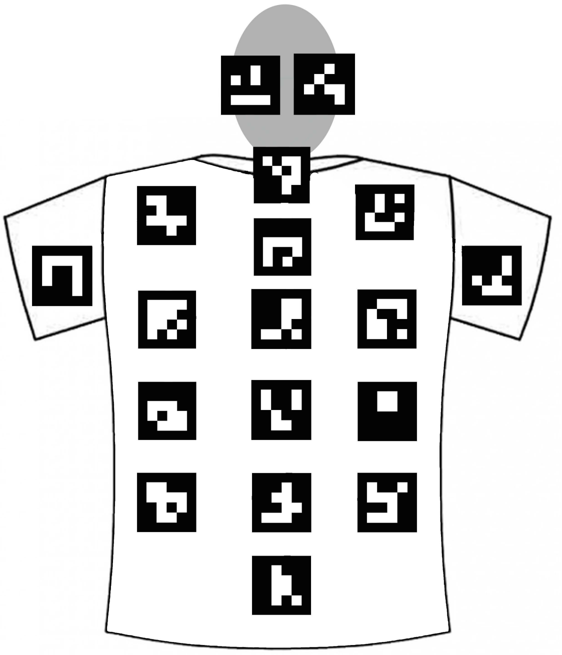


Figure 3.2: Adopted markers arrangement

marker was printed on regular paper having a surface of $(4.3 \times 4.3)cm^2$ and then applied onto a white plastic rigid plane surface having a surface of $(5 \times 5)cm^2$. This choice lets the markers be surrounded by a thin white plastic edge which helps the detection of the markers themselves. In order to attach the markers to the t-shirt some Velcro was sewn on the back of the t-shirt and stuck on the back of the markers plastic supports. The final result of these preliminary operations is displayed in the figure 3.3.

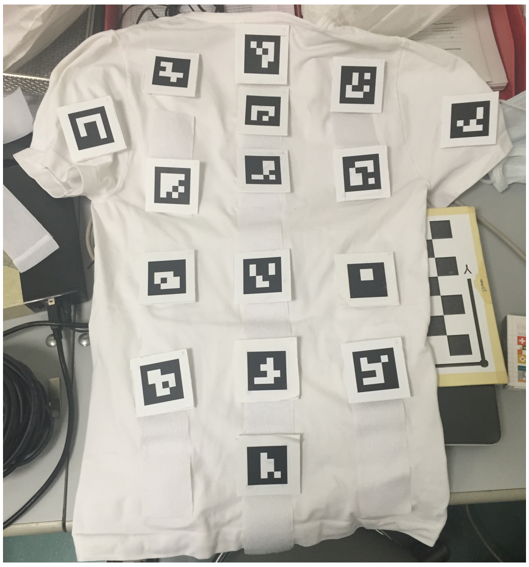


Figure 3.3: A picture of the t-shirt used in the operations

The total number of markers in use is then eighteen: sixteen for the back, the shoulders and the neck and two for the head. This amount was considered suitable for the study since it allows a good representation of the upper body posture without making the t-shirt too heavy or uncomfortable for the monitored operator.

3.3 Postural assessment criteria

During the past fifty years several techniques for assessing physical exposure to work-related musculoskeletal risks have been developed, with particular emphasis on posture-based methods. The options for postural assessment methodologies are then various: videotaping and computer-aided analysis, direct or instrumental techniques, observation-based methods by pen and paper and various approaches for self-reporting assessment. Advantages and disadvantages of each method have been highlighted and discussed in the years [23]. The assessment criterion we decided to adopt is *Rapid Upper Limb Assessment*, also known as *RULA*, which is one of the most employed methods nowadays.

3.3.1 Rapid Upper Limb Assessment (RULA)

The Rapid Upper Limb Assessment (RULA) method has been developed by Dr.Lynn McAtamney and Professor E. Nigel Corlett, ergonomists from the University of Nottingham, England (Dr. McAtamney is now at Telstra, Australia) [24]. RULA is a postural targeting method for estimating the risks of workrelated upper limb disorders. A RULA assessment gives a quick and systematic assessment of the postural and musculoskeletal risks for the examined subject.

The main characteristic of this assessment method is that it uses numbers to represent postures with an associated coding system. In order to compute the total RULA score associated with a certain posture, it is necessary to analyze the different body parts separately, following the procedures step and assigning to each one of them a partial score which eventually contributes to compute a final score whose value qualitatively classifies uncorrectness and risk for such a posture.

The score assigned to a specific posture basically depend on three factors:

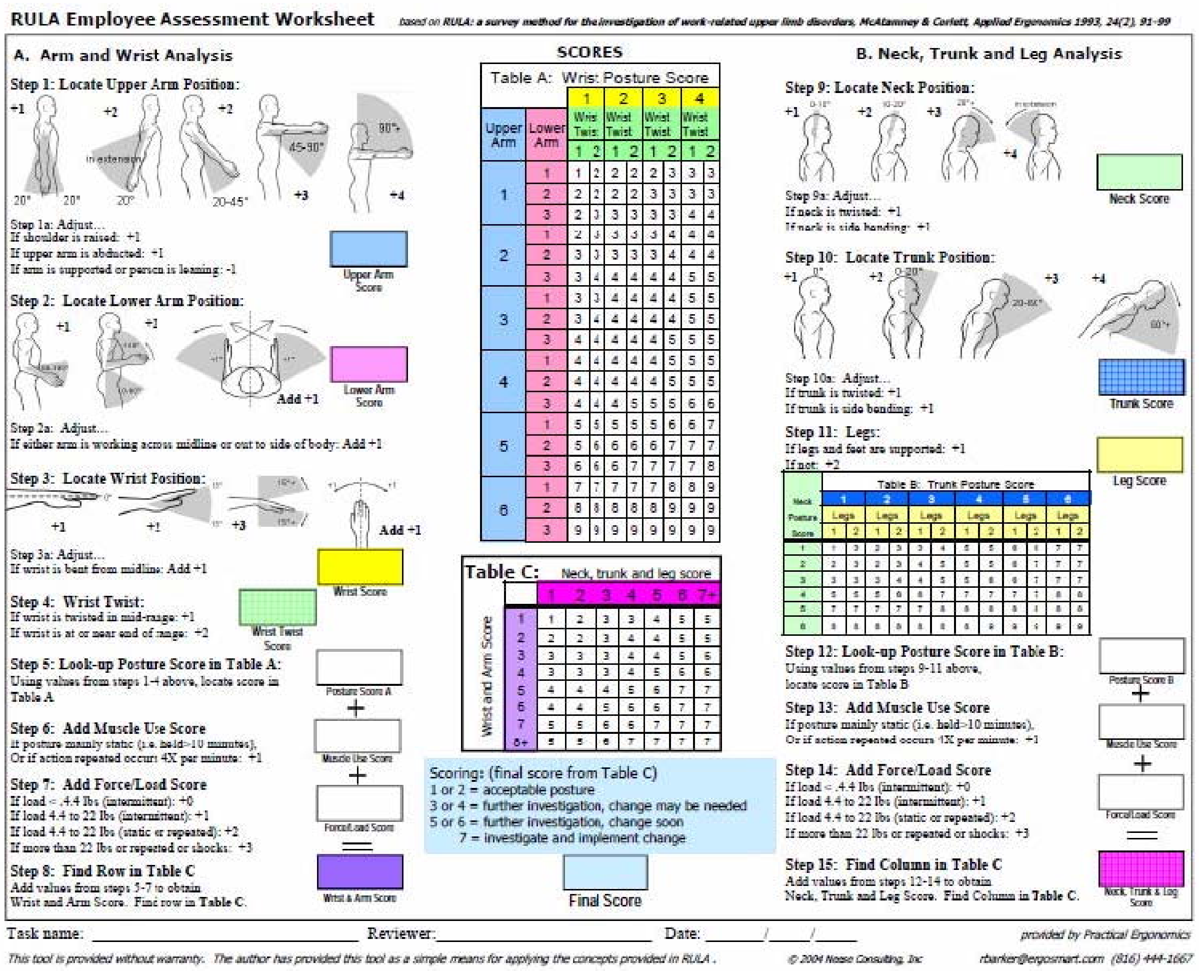
1. naturalness and ease of the position assumed;

2. time spent in the examined position;

3. load to be sustained during the position;

The traditional RULA algorithm uses body posture diagrams and three different scoring tables: the first one (table A in figure 3.4) is for wrist posture score with respect to upper and lower arm position; the second one is for neck, trunk and leg score with respect to wrist and arm position (table C in figure 3.4); the third one is for trunk posture score with to neck position (table B in figure 3.4). The human body is divided in two main groups: one for arm and wrist analysis (left side in the figure) and the other one for neck, trunk and legs analysis (right side in the figure). A scoring system is used to assign scores at every subgroup region in order to compute a final total score for the posture. Scores are expressed with positive integer values starting from +1: the lowest the score, the most correct is the assumed position.

As we mentioned before, in addition to the analysis of the posture recordings, there are several other risk factors taken into account by RULA for assessing the quality of a posture. In particular, force exertion,



RULA Employee Assessment Worksheet *based on RULA: a survey method for the investigation of work-related upper limb disorders, McAtamney & Corlett, Applied Ergonomics 1993, 24(2), 91-99*

A. Arm and Wrist Analysis

Step 1: Locate Upper Arm Position:

+1
+2
+2
+3
+4
20°
20°
in extension
20°
20-45°
45-90°
90°+

Step 1a: Adjust…
If shoulder is raised: +1
If upper arm is abducted: +1
If arm is supported or person is leaning: -1

Upper Arm Score

Step 2: Locate Lower Arm Position:

Lower Arm Score

Step 2a: Adjust…
If either arm is working across midline or out to side of body: Add +1

Step 3: Locate Wrist Position:

Wrist Score

Step 3a: Adjust…
If wrist is bent from midline: Add +1

Step 4: Wrist Twist:
If wrist is twisted in mid-range: +1
If wrist is at or near end of range: +2

Wrist Twist Score

Step 5: Look-up Posture Score in Table A:
Using values from steps 1-4 above, locate score in Table A

Posture Score A
+

Step 6: Add Muscle Use Score
If posture mainly static (i.e. held>10 minutes),
Or if action repeated occurs 4X per minute: +1

Muscle Use Score
+

Step 7: Add Force/Load Score
If load < 4.4 lbs (intermittent): +0
If load 4.4 to 22 lbs (intermittent): +1
If load 4.4 to 22 lbs (static or repeated): +2
If more than 22 lbs or repeated or shocks: +3

Force/Load Score
=

Step 8: Find Row in Table C
Add values from steps 5-7 to obtain
Wrist and Arm Score. Find row in Table C.

Wrist & Arm Score

SCORES

Table A: Wrist Posture Score

		1		2		3		4	
Upper Arm	Lower Arm	Wrist Twist		Wrist Twist		Wrist Twist		Wrist Twist	
		1	2	1	2	1	2	1	2
1	1	1	2	2	2	2	3	3	3
	2	2	2	2	2	3	3	3	3
	3	2	3	3	3	3	3	4	4
2	1	2	3	3	3	3	4	4	4
	2	3	3	3	3	3	4	4	4
	3	3	4	4	4	4	4	5	5
3	1	3	3	4	4	4	4	5	5
	2	3	4	4	4	4	4	5	5
	3	4	4	4	4	4	5	5	5
4	1	4	4	4	4	4	5	5	5
	2	4	4	4	4	4	5	5	5
	3	4	4	4	5	5	5	6	6
5	1	5	5	5	5	5	6	6	7
	2	5	6	6	6	6	7	7	7
	3	6	6	6	7	7	7	7	8
6	1	7	7	7	7	7	8	8	9
	2	8	8	8	8	8	9	9	9
	3	9	9	9	9	9	9	9	9

Table C: Neck, trunk and leg score

Wrist and Arm Score	1	2	3	4	5	6	7+
1	1	2	3	3	4	5	5
2	2	2	3	4	4	5	5
3	3	3	3	4	4	5	6
4	3	3	3	4	5	6	6
5	4	4	4	5	6	7	7
6	4	4	5	6	6	7	7
7	5	5	6	6	7	7	7
8+	5	5	6	7	7	7	7

Scoring: (final score from Table C)
1 or 2 = acceptable posture
3 or 4 = further investigation, change may be needed
5 or 6 = further investigation, change soon
7 = investigate and implement change

Final Score

B. Neck, Trunk and Leg Analysis

Step 9: Locate Neck Position:

Neck Score

Step 9a: Adjust…
If neck is twisted: +1
If neck is side bending: +1

Step 10: Locate Trunk Position:

Trunk Score

Step 10a: Adjust…
If trunk is twisted: +1
If trunk is side bending: +1

Step 11: Legs:
If legs and feet are supported: +1
If not: +2

Leg Score

Table B: Trunk Posture Score

Neck Posture Score	1		2		3		4		5		6	
	Legs		Legs		Legs		Legs		Legs		Legs	
	1	2	1	2	1	2	1	2	1	2	1	2
1	1	3	2	3	3	4	5	5	6	6	7	7
2	2	3	2	3	4	5	5	5	6	7	7	7
3	3	3	3	4	4	5	5	6	6	7	7	7
4	5	5	5	6	6	7	7	7	7	7	8	8
5	7	7	7	7	7	8	8	8	8	8	8	8
6	8	8	8	8	8	8	8	9	9	9	9	9

Step 12: Look-up Posture Score in Table B:
Using values from steps 9-11 above,
locate score in Table B

Posture Score B
+

Step 13: Add Muscle Use Score
If posture mainly static (i.e. held>10 minutes),
Or if action repeated occurs 4X per minute: +1

Muscle Use Score
+

Step 14: Add Force/Load Score
If load < 4.4 lbs (intermittent): +0
If load 4.4 to 22 lbs (intermittent): +1
If load 4.4 to 22 lbs (static or repeated): +2
If more than 22 lbs or repeated or shocks: +3

Force/Load Score
=

Step 15: Find Column in Table C
Add values from steps 12-14 to obtain
Neck, Trunk and Leg Score. Find Column in Table C.

Neck, Trunk & Leg Score

Task name: ______________ Reviewer: ______________ Date: ____/____/____

provided by Practical Ergonomics

This tool is provided without warranty. The author has provided this tool as a simple means for applying the concepts provided in RULA.

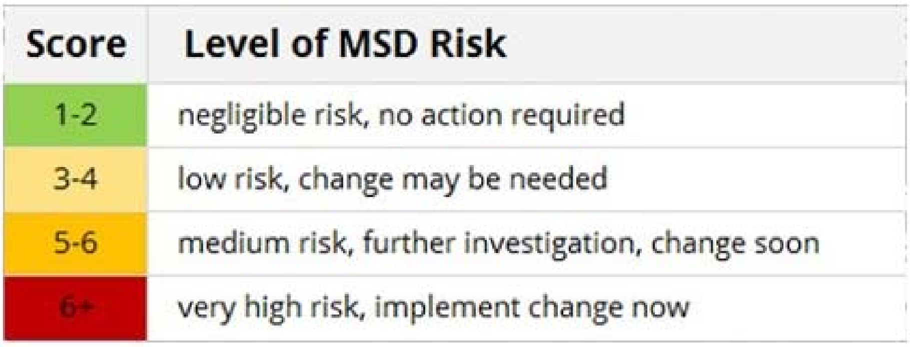
© 2004 Neese Consulting, Inc

rbarker@ergosmart.com (816) 444-1667

Figure 3.4: Example of a RULA worksheet

repetition (or frequency) of the same movements and time spent in the same position. A score is given for muscle employment by evaluating how frequently the same posture is repeated and it is expressed in occurrences per minute. Another score is attributed to the load to be sustained: it depends not only on the amount of load but also on how long it is kept; for example, if the weight is less than two kilograms and it is held intermittently, the score is zero.

Once all the individual scores related to the different upper body regions are determined, the final RULA score can be computed following its algorithm. According to the final value obtained, it is possible to indicate if it is necessary to intervene and the level of intervention required to reduce the risk of MSDs for the operator.



Score	Level of MSD Risk
1-2	negligible risk, no action required
3-4	low risk, change may be needed
5-6	medium risk, further investigation, change soon
6+	very high risk, implement change now

Figure 3.5: Levels of MSD risk according to RULA scores

The reason behind the choice of RULA can be listing pros and cons. Its positive qualities can be summarized in the following point:

- RULA is ideal for sedentary workers;
- It is influential in the ergonomics field and it does not require special equipment;
- It is quick and easy to complete even without an advanced knowledge in ergonomics;
- RULA scores can already give a general indication of the level of intervention to be applied for reducing MSD risks.

On the other hand, RULA presents the following limitations:

- It does not include an assessment for fingers and thumb posture;
- It doesn't provide an integrated assessment of all the biomechanical risk factors;

The method has been tested in a laboratory situations relative to operations involving Visual Display Units [24] but its sensitivity, specificity and predictive value for quantifying the actual risk for musculoskeletal injuries has not been assessed yet.

This work is not meant for proposing a new rule for determining ergonomics, it will focus on calculating and tracking head, neck, back and shoulders posture in order to apply RULA criteria. For computing the score of the remaining subregions examined by RULA algorithm, such as upper and lower arm, wrist and legs they will be visually estimated by looking at the frames but without applying a technical method to obtain exact position values.

3.3.2 Novel Ergonomic Postural Assessment (NERPA)

The *Novel Ergonomic Postural Assessment Method (NERPA)*, as the name suggests, is a postural assessment method designed for product processes [25], mainly developed for industrial manual assembly tasks, frequent in the automotive industry. It is developed with the help of a digital human model by a 3D CAD tool. This method has some common points with the RULA method (still the most commonly used in industrial environments), as for example it maintains the original A,B,C tables but it also introduces significant differences for the arms, neck, trunk and wrists. The angular values of each body region are modified, by updating RULA method using the most congruous standards. The method development is centered on using a Digital Human Model (DHM), integrated with a 3D product-process environment design in order to assess worker posture and measure the level of risk.

In order to conduct the ergonomic evaluations in a simulation environment, it is necessary to define all the resources, including the 3D geometry of the workstation, a DHM that can cover the range of the involved population, the 3D geometry of the assembly of parts, and a definition of all assembly tasks. NERPA method is based on five phases:

1. Collection of workers opinions and records of injuries;

2. Postural analysis implementation by using RULA and NERPA methods:

3. Evaluation of the possible ergonomic improvements in the modeled environment defined before;

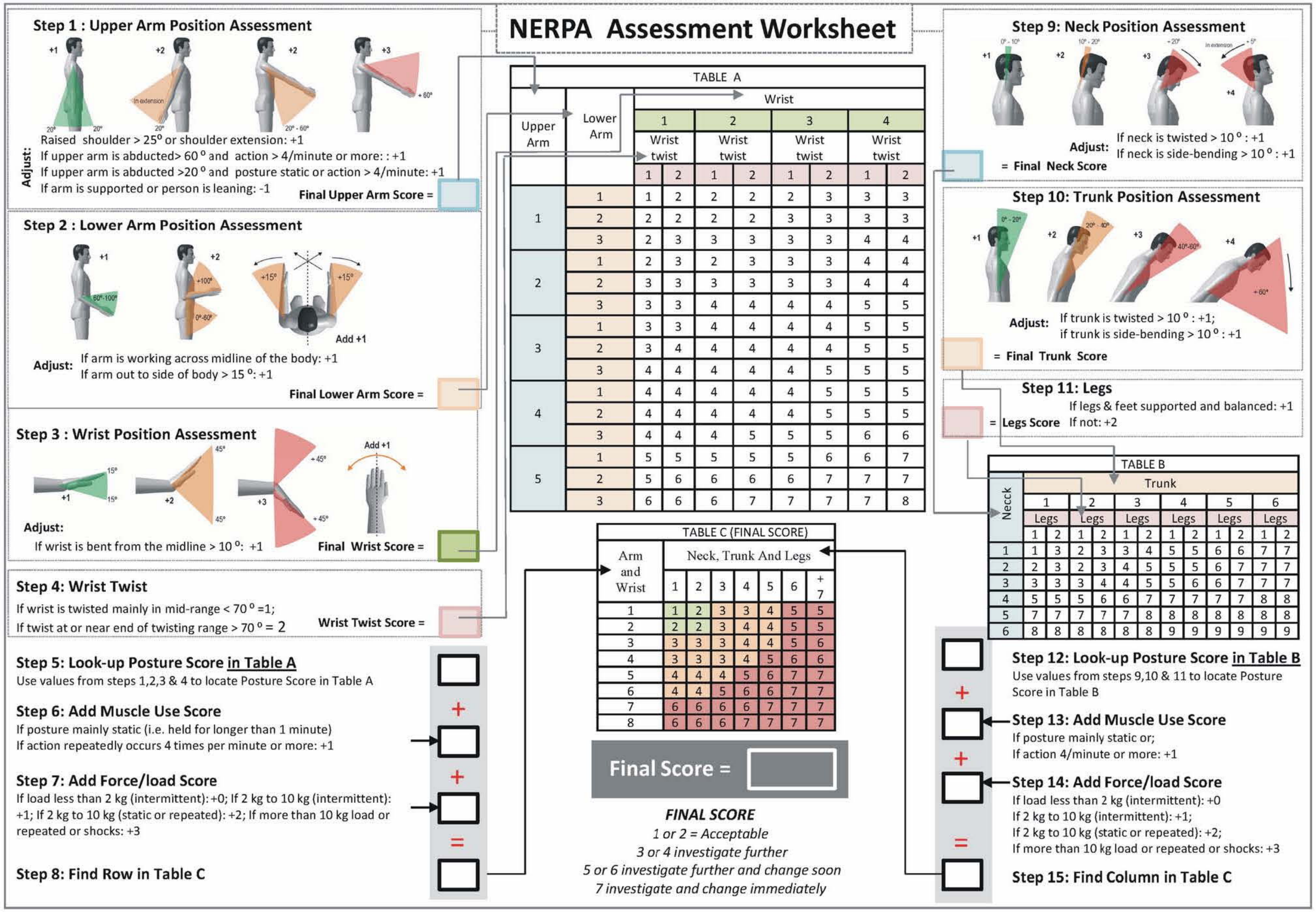
4. Proposed improvements implementation by using a tracking system;

5. Evaluation of the improvements in reals workstations;

One of the aspects that make NERPA differ from RULA is a different upper arm posture assessment. In fact, NERPA considers three (instead of RULA's four) bending position levels for the arm. This way the angle intervals covered by the levels increase in size avoiding unnecessary penalization for some common working postures that do not constitute, a priori, any risks for the worker. In figure 3.6 the NERPA evaluation table is displayed.

Also trunk and neck posture assessments differ from one method to the other. In fact, when considering trunk inclinations, the angles intervals are differently classified; in particular: the first one is extended by ten degrees, the second one by twenty degrees and the last one's lower limit is ten degrees higher. For what concerns the neck, NERPA penalizes neck torsion and side-inclination only if they result higher than a given threshold.If the neck experiences a torsion or side-inclination above ten degrees wide, then the neck final score has to be increased by one. On the other hand, neck frontal bending values are computed as in RULA algorithm. Lastly, similarly to what it does for the neck posture evaluation, NERPA allows a fifteen degrees wide wrist inclination before penalization, differently from RULA.

It has been proved [25] that NERPA can obtain a sixteen percent accuracy improvement with respect to RULA. This is primarily due to its implementation through a 3D simulation tool which allows angle observation and angle and final posture evaluation via software.



NERPA Assessment Worksheet

Step 1 : Upper Arm Position Assessment

Adjust:
Raised shoulder > 25° or shoulder extension: +1
If upper arm is abducted> 60 ° and action > 4/minute or more: : +1
If upper arm is abducted >20 ° and posture static or action > 4/minute: +1
If arm is supported or person is leaning: -1

Final Upper Arm Score =

Step 2 : Lower Arm Position Assessment

Adjust: If arm is working across midline of the body: +1
If arm out to side of body > 15 °: +1

Final Lower Arm Score =

Step 3 : Wrist Position Assessment

Adjust:
If wrist is bent from the midline > 10 °: +1

Final Wrist Score =

Step 4: Wrist Twist
If wrist is twisted mainly in mid-range < 70 ° =1;
If twist at or near end of twisting range > 70 ° = 2

Wrist Twist Score =

Step 5: Look-up Posture Score in Table A
Use values from steps 1,2,3 & 4 to locate Posture Score in Table A

Step 6: Add Muscle Use Score
If posture mainly static (i.e. held for longer than 1 minute)
If action repeatedly occurs 4 times per minute or more: +1

Step 7: Add Force/load Score
If load less than 2 kg (intermittent): +0; If 2 kg to 10 kg (intermittent): +1; If 2 kg to 10 kg (static or repeated): +2; If more than 10 kg load or repeated or shocks: +3

Step 8: Find Row in Table C

TABLE A

Upper Arm	Lower Arm	Wrist							
		1		2		3		4	
		Wrist twist		Wrist twist		Wrist twist		Wrist twist	
		1	2	1	2	1	2	1	2
1	1	1	2	2	2	2	3	3	3
	2	2	2	2	2	3	3	3	3
	3	2	3	3	3	3	3	4	4
2	1	2	3	2	3	3	3	4	4
	2	3	3	3	3	3	3	4	4
	3	3	3	4	4	4	4	5	5
3	1	3	3	4	4	4	4	5	5
	2	3	4	4	4	4	4	5	5
	3	4	4	4	4	4	5	5	5
4	1	4	4	4	4	4	5	5	5
	2	4	4	4	4	4	5	5	5
	3	4	4	4	5	5	5	6	6
5	1	5	5	5	5	5	6	6	7
	2	5	6	6	6	6	7	7	7
	3	6	6	6	7	7	7	7	8

TABLE C (FINAL SCORE)

Arm and Wrist	Neck, Trunk And Legs						
	1	2	3	4	5	6	+ 7
1	1	2	3	3	4	5	5
2	2	2	3	4	4	5	5
3	3	3	3	4	4	5	6
4	3	3	3	4	5	6	6
5	4	4	4	5	6	7	7
6	4	4	5	6	6	7	7
7	6	6	6	6	7	7	7
8	6	6	6	7	7	7	7

Final Score =

FINAL SCORE
1 or 2 = Acceptable
3 or 4 investigate further
5 or 6 investigate further and change soon
7 investigate and change immediately

Step 9: Neck Position Assessment

Adjust: If neck is twisted > 10 ° : +1
If neck is side-bending > 10 ° : +1

= Final Neck Score

Step 10: Trunk Position Assessment

Adjust: If trunk is twisted > 10 ° : +1;
if trunk is side-bending > 10 ° : +1

= Final Trunk Score

Step 11: Legs
If legs & feet supported and balanced: +1
= Legs Score If not: +2

TABLE B

Necck	Trunk											
	1		2		3		4		5		6	
	Legs		Legs		Legs		Legs		Legs		Legs	
	1	2	1	2	1	2	1	2	1	2	1	2
1	1	3	2	3	3	4	5	5	6	6	7	7
2	2	3	2	3	4	5	5	5	6	7	7	7
3	3	3	3	4	4	5	5	6	6	7	7	7
4	5	5	5	6	6	7	7	7	7	7	8	8
5	7	7	7	7	7	8	8	8	8	8	8	8
6	8	8	8	8	8	8	9	9	9	9	9	9

Step 12: Look-up Posture Score in Table B
Use values from steps 9,10 & 11 to locate Posture Score in Table B

Step 13: Add Muscle Use Score
If posture mainly static or;
If action 4/minute or more: +1

Step 14: Add Force/load Score
If load less than 2 kg (intermittent): +0
If 2 kg to 10 kg (intermittent): +1;
If 2 kg to 10 kg (static or repeated): +2;
If more than 10 kg load or repeated or shocks: +3

Step 15: Find Column in Table C

Figure 3.6: Example of a NERPA assessment worksheet

4
Model and methodology

In this chapter the followed model, its implementation and all the technical steps necessary to make it properly work are presented. In the first section we present the full setup of the operating room and a description of the hardware used. Then the second section describes the actual detection phase of the process, with an explanation of the necessary steps for the software to acquire the relevant data. Lastly, the analysis phase is discussed in the third and last section, which also includes a description of the techniques employed for analyzing the collected data.

4.1 Experimental setup

The main component of our acquisition setup consists in a two-cameras system used for capturing the frames from which data will be detected. The cameras adopted are mono-colour C-Mount CMOS digital cameras and their resolution is 2592 × 2048 pixels. We adopted a 12 millimeters focal length on the cameras that, despite giving good and readable results, did not seem to be the optimal choice as it did introduce a bit of vignetting in the corners of the images. The cameras were installed to two tripods, displaced at a height of approximately one meter from the floor and at a distance of approximately 60 centimeters from the markers if the an operation was on the left side of the patient, while at a distance between 1 and 1.5 meters, if the operation 1,5 meter if the record was on the right side. This difference was due to room space limitations. A typical operating room arrangement is shown in figures 4.1 and 4.2.

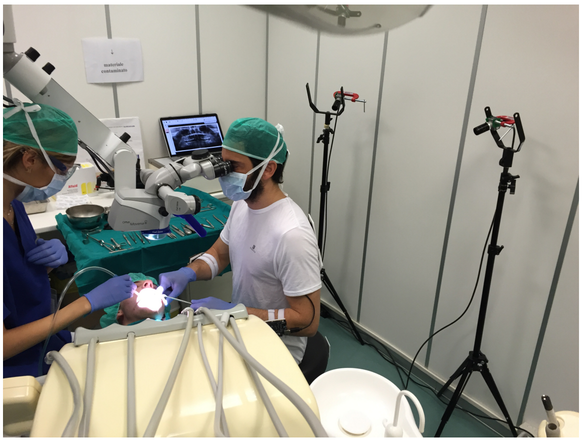


Figure 4.1: Cameras disposition for a left side tooth removal operation

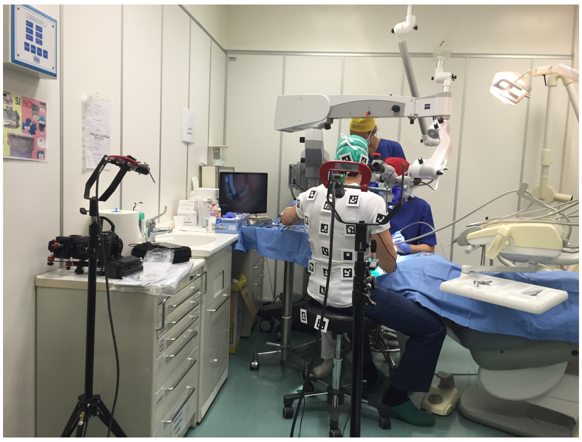


Figure 4.2: Cameras disposition for a right side tooth removal operation

The two cameras in use are powered by *Power Over Ethernet (POE)* and for connecting them to our laptop we used a POE Gigabit switch. In order to save all the frames recorded by the camera we used a notebook with an Intel i7 processor. Then the frames were stored on a external hard disk drive connected to the computer by a 3.0 USB connection.

Acquisition rate was not kept constant through all operations but it was always approximately about 1 frame per second and it was precisely recorded at every operation. We accepted this range of frame rate (which could be considered quite low in other posture assessment applications) because it was high enough to record the typical kind of movements of an operating dentist as they are usually circumscribed and slow

A typical frame captured by the camera system and recorded on the computer displays a posterior view of the operator and of the marker-equipped white t-shirt worn by him during an operation. An example of a recorded frame is shown in figure 4.3

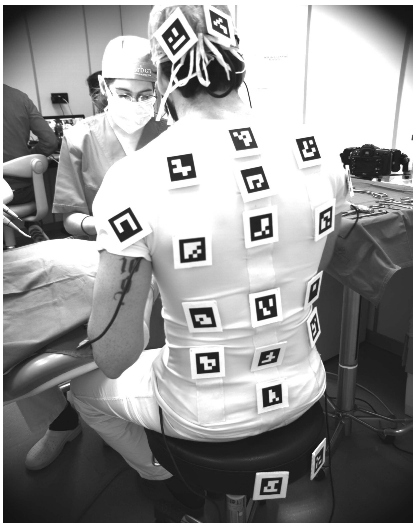


Figure 4.3: Example of a recorded frame

After the acquisition of the markers, for every single operation recorded, data for calibration were acquired by capturing pictures of the chessboard image used as planar calibration target. These data were stored in order to be fed to the OpenCV calibration utility [26].

4.2 Detection phase

In order to track the markers, we adopted the ArUco library [13] which is a minimal library for augmented reality applications based on OpenCV. The marker detection process of ArUco can be schematized as follows:

1. An adaptive thresholding algorithm is applied for detecting markers borders;
2. All contours get detected and, therefore, markers are detected,too. However, now the image needs to be filtered in order to remover unnecessary detected borders (part 1 of figure 4.4).
3. All the borders outside the area of interest and those characterized by a too small number of points are removed (part 2 of figure 4.4);

4. Detected contours go through a polygonal approximation procedure and only concave contours with exactly four corners are considered (part 3 of figure 4.4);

5. Corners are sorted following an anti-clockwise order;

6. Rectangles that result being too close to each other (according to a given threshold) are detected and some of them are removed. This is necessary because the adaptive threshold detects both internal and external borders of the markers and the external ones are the only ones relevant for our data analysis (part 4 of figure 4.4);

7. Markers are identified;

8. Projection perspective is reverse so as to obtain a frontal view of the markers rectangular area. This can be done thanks to the projection homography detected in the calibration phase(part 5 of figure 4.4);

9. The markers areas are segmented and processed through Otsu's algorithm [27]. This algorithm assumes a bimodal distribution of white and black pixel values and it finds the threshold that maximizes the extra-class variance and minimizes the the intra-class one (part 6 of figure 4.4);

10. Internal code of markers are identified. Every marker has its own internal code. Markers are divided in a 6×6 grid where only the internal 5×5 group of cells contains id information. This is the first time that the presence of the black borders of the markers is checked. The internal cells are read by the software algorithm which checks if they provide a valid code. Sometimes rotations of the detected markers are necessary before reading the code;

11. For all the markers correctly identified corners are refined through a subpixel interpolation process;

12. Provided camera parameters had already been extracted and fed to the algorithm, the extrinsic parameters of the markers are computed.

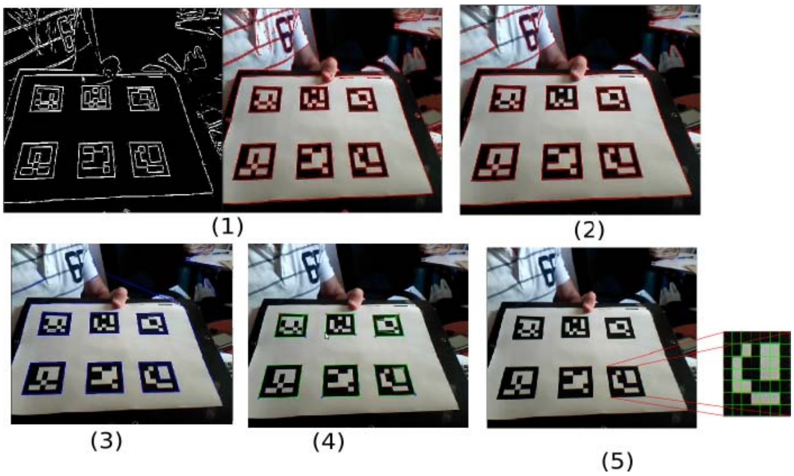


Figure 4.4: ArUco markers detection steps

4.2.1 Marker detection

The adopted ArUco version is structured in four phases:

1. Detection of marker candidates;
2. Candidate codification check (markers identification);
3. Detected markers filtering;
4. Corner subpixel refinement;

The detection of the 3D coordinates of the marker is possible because, knowing the camera intrinsic parameters (camera matrix and distortion coefficients obtained by calibration) and specifying the size of the markers, the library can return the positions of the markers with respect to the camera. ArUco uses the class named Marker characterized by the following features: a vector of four 2D points representing the corners of the image, its size expressed in meters, the translation and rotation coordinates that connects the center of the marker and the camera system origin. The 3D mapping information is stored in these last values stored into variables *Tvec* and *Rvec*, respectively storing values for 3D translation and for 3D rotation transform expressed as given by the Rodrigues formula in a 3×3 matrix.

In order to project a 3D point from a reference system centered on a marker onto the camera reference system, its homogeneous coordinates vector has to be multiplied by the proper transformation matrix whose values can be computed from the Tvec and Rvec values. The resulting

3D point can then be mapped into world space by conveniently applying the camera matrix. Finally it can be processed for compensating the image according to the estimated distortion model.

The result of the whole detection procedure is that of obtaining the position of the markers, stored in the Rvec and Tvec variables. In our software implementation these values are saved in a text file in order to allow an easier reuse in the post-detection analysis phase.

4.2.2 3D coordinates computation

The computation of the 3D coordinates of the markers is accomplished by a different script.The values previously stored in the text file are loaded in this new script and, using the Rodrigues formula [28], it is possible to compute the rotation matrix and the translation vector necessary for the 3D position of the markers which can then be computed according to the following formula:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{t} \tag{4.1}$$

In our implementation the script outputs two different text files: in the first one it saves the 3D positions and the t translation vector of each marker, in the second one it saves the **R** matrix and the **t** translation vector of each marker.

4.3 Analysis phase

The goal of the analysis phase is that of computing as much useful information as possible from the data obtained in the detection phase. In order to process the data we used Matlab computing environment which allowed us not only to compute useful information for postural assessment but also to represent them through visually direct representations such as the motion history reconstruction of the markers and the 3D skeleton/body reconstruction.

4.3.1 Motion history

The motion history image method (MHI) [29] [30] is a technique which allows representing the markers motion sequence in a compact manner, by tracing their path through their inter-frames positions differences.

The Matlab script used for computing the motion history graph of the operations monitored loads the 2D coordinates of the corner points of the markers from the stored data text file, it then plots the edges of the markers and colors the markers area with a soft blue shadow. The more the marker square superimpose on each other, the more intense will be the blue color in the common location. In order to obtain an equal map of different blue intensity values for operations having completely different time durations, a normalization step with respect to operation durations is applied. Another important information that can be read from MHIs is about how much displacement was achieved throughout the operation with respect to the starting reference proper posture. Hence, the algorithm also displays the reference markers positions by highlighting their edges in red color. An example of MHI obtained through our script is displayed in figure 4.5.

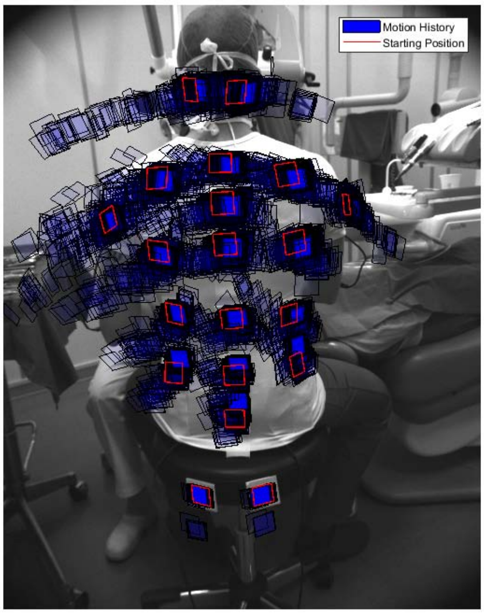


Figure 4.5: Example of motion history image

4.3.2 Skeleton 3D reconstruction

The skeleton 3D reconstruction is a process of key importance for the whole analysis phase as many other analysis steps are based on its results. The employed algorithm allows a faithful reconstruction of the body posture of the dentist through the entire surgical operation. The relative scripts loads the stored 3D coordinate of the markers and is aided by an interpolation procedure to estimate the bottom neck marker (identification code 0) for those frames in which it is not properly captured by the cameras. This can happen due to some random external light flashes on the marker. We adopted this method because, without the information of this key marker, we could not connect the head to the body in the 3D reconstruction. At the end of the script the values of the positions of the markers in space are plotted and, by connecting the points, a 3D skeleton reconstruction is built. In order to have a better graphical representation, we decidef to assign different colors to to the different body parts connections: the "neck" (namely the connection between head and body) is represented in green, the backbone in yellow, the connection between neck and shoulders in black and the marker lines parallel to the central one and their connection to the spinal column are represented in two tones of red and blue. A scheme of the color code adopted is represented in figure 4.6.

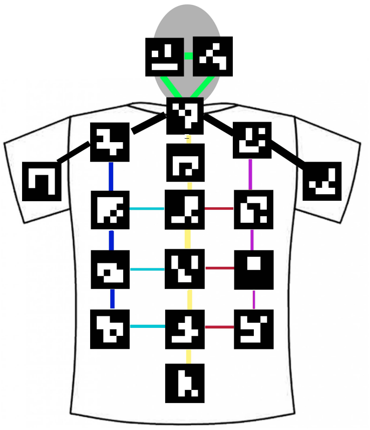


Figure 4.6: Scheme of the connections and the colors used to represent them in the 3D reconstruction algorithm

It is also possible to plot on the same 3D reconstruction image the vectors representing the markers coordinates systems axes. In figure 4.7

a sample image produced by the 3D reconstruction script is shown with the orange lines representing the direction of the z axis for each marker reference system.

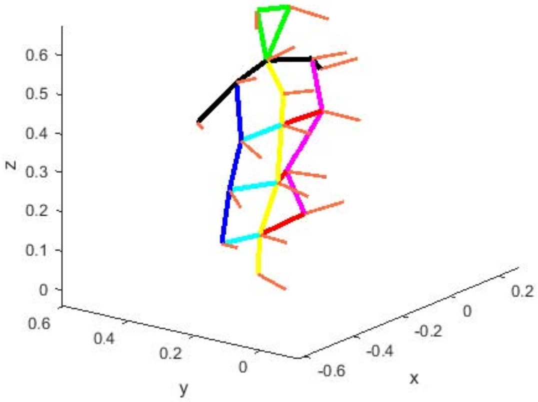


Figure 4.7: Example of 3D skeleton reconstruction

4.3.3 Angular reconstruction

Angular reconstruction is an important data analysis step necessary for computing the final RULA scores for evaluating posture during the dental operations. Our study focuses on the upper part of the body, with the goal of locating neck an trunk positions. In order to obtain their angles, the angular reconstruction script stars by loading the text file with the pre-computed values of the 3D positions of the markers.The method we decided to adopt to calculate the angles is based on the projection of the 3D points on a sagittal plane passing through spinal column. In order to estimate the parameters of the sagittal plan we computed a matrix **X** with the position coordinates of the spinal column markers, then we computed its mean value for computing a new matrix **Y** with the differences between the **X** values and the mean. These preliminary steps are needed to compute the **Z** matrix, given by the following formula:

$$\mathbf{Z} = \mathbf{Y'} * \mathbf{Y} \tag{4.2}$$

We then computed its eigenvalues. Given the equation of a general plan as:

$$ax + by + cz + d = 0 \tag{4.3}$$

where a, b and c are the elements of the eigenvector associated with the smallest eigenvalue of $\mathbf{Z}$ matrix. In order to evaluate the d parameter we imposed the passage through the mean point by substituting its coordinate in the equation. The estimate plate is then represented in 3D plot; an example is given in figure 4.8.

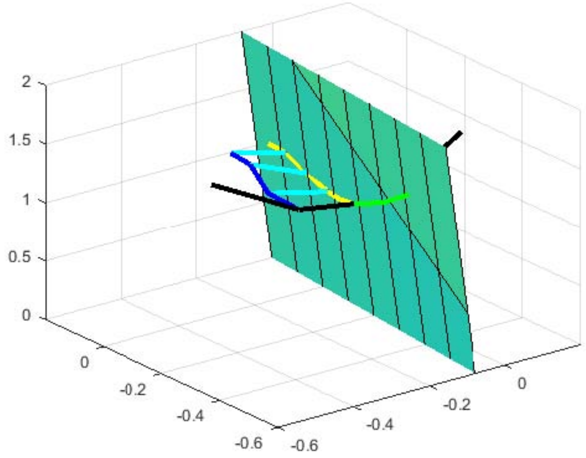


Figure 4.8: Representation of the sagittal plane crossing a 3D skeleton reconstruction

Once the sagittal plane parameters were computed, the coordinates of the projections of the markers points onto the sagittal plane are computed according to the following formulas:

$$P_{sagittal,id,x} = (1 - a^2)x_p - aby_p - acz_p - ad \tag{4.4}$$

$$P_{sagittal,id,y} = (1 - b^2)y_p - abx_p - bcz_p - bd \tag{4.5}$$

$$P_{sagittal,id,z} = (1 - c^2)z_p - acx_p - bcy_p - cd \tag{4.6}$$

where x_p, y_p, z_p are the coordinates of the id marker point. This last step is useful for evaluating the angle between the vectors connecting the relevant points on the sagittal plane. Through this computation the inclination angles expressed in terms of the camera reference system are computed.

In order to estimate the inclination angle of the neck the sagittal plane passing through the centers of marker 13 (one of the two head markers), marker 0 (neck bottom marker) and one point having same x and y coordinates as marker 0 center and same z coordinate as marker 13 center was computed. The coordinates of these points were then used for computing the angle between the vectors connecting the center of marker 0 with the other two points. The computed value finally was adjusted to take in considerations the contribution given to it by the trunk inclination.

Among the other relevant factors for RULA there are trunk's side-bending, twist and inclination angles. The procedure to evaluate trunk's side-bending angle is similar to that just described: the sagittal plane passing through the centers of markers 9, 6 and 1, respectively the two on the lower part of the trunk and the one between shoulder blades. We then projected the central points of markers 7 and 4 (corresponding to the shoulders markers) and computed the angle between the vector connecting marker 9 with marker 6 and the vector connecting the projections of marker 7 and marker 4. If the angle is close to 0 degrees, RULA trunk's side-bending score is null. In particular the trunk's side-bending score is attributed only when this angle is wider than 7 degrees (considering both possible directions).

In order to evaluate trunk's twist we exploited the previously computed plane and measured the angles between the vector connecting marker 7 and marker 4 central points and the normal vector of the plane. If the estimated angle is close to 90 degrees, the trunk's twist RULA score is null. In particular the trunk's twist score is attributed only when this angle is bigger or smaller than 90 degrees by at least 5 degrees.

Finally, in order to estimate trunk's inclination a procedure similar to the one adopted for neck's inclination. The sagittal plane passing through the central points of markers 16 and 0 and through one point having same x and y coordinates as marker 16 center and same z coordinate as marker 0 center was determined. The angle between the projections of the vectors connecting marker 16 with the other two points onto the sagittal plane was then computed and used as trunk's inclination angle.

5 Experimental results

In this chapter we present the results obtained from the analyses by the models presented in the previous chapter. These results are the consequence of an analysis of a sample based on 60 surgical operations: 20 using microscope, 20 using loupes and 20 with naked eye. In the first section we show the postural comparison of the three surgical techniques using the motion history analysis with respect to the desired position. In the second section we present the values of the angles regarding the trunk and the head. In the third and last section we evaluated our results according to RULA scoring system

5.1 Motion history images comparison

The technique described in previous chapter for estimating the motion history image of an entire operation gives us a good element of comparison between the different visual aids employed.

Microscope

Figure 5.1 displays a left side operation performed with the aid of a microscope. The motion history analysis shows that the head and the trunk are quite static if compared with the results of the other two methods, as their displacements result shorter. The static is evident because in the left picture we have a intense blue in some positions and this element means that the marker has been captured in that position multiple times. This result is visible also in figure 5.2, displaying a microscope-aided right side operation: in the right part of the figure the yellow isoline indicates a high number of marker occurrences upon those points, almost all coinciding with the reference posture markers positions, therefore pointing out that the dentist has frequently kept a similar posture during the operation.

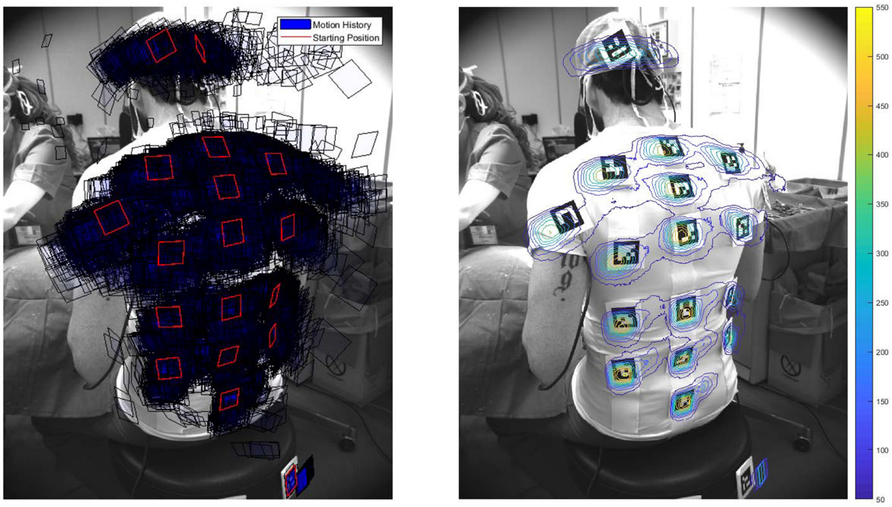


Figure 5.1: left: MHI in a left side microscope-aided operation; right: corresponding colormap for number of overlapping markers position

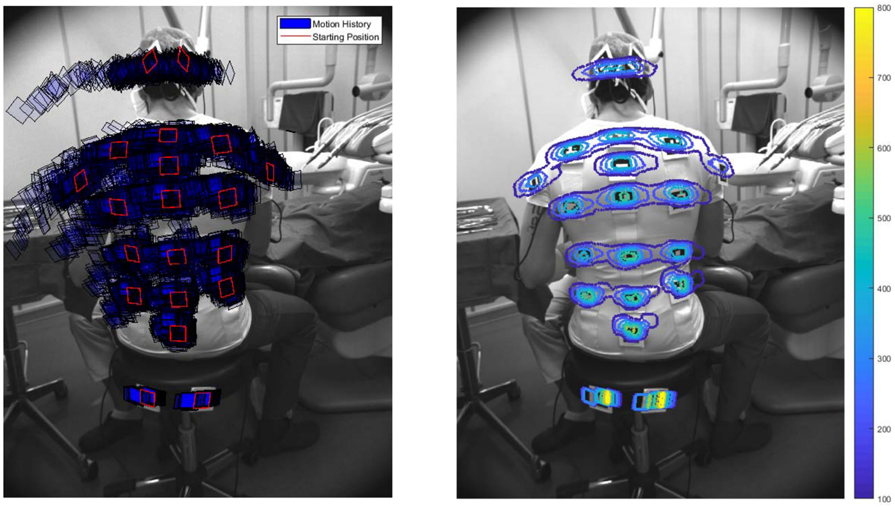


Figure 5.2: left: MHI in a right side microscope-aided operation; right: corresponding colormap for number of overlapping markers position

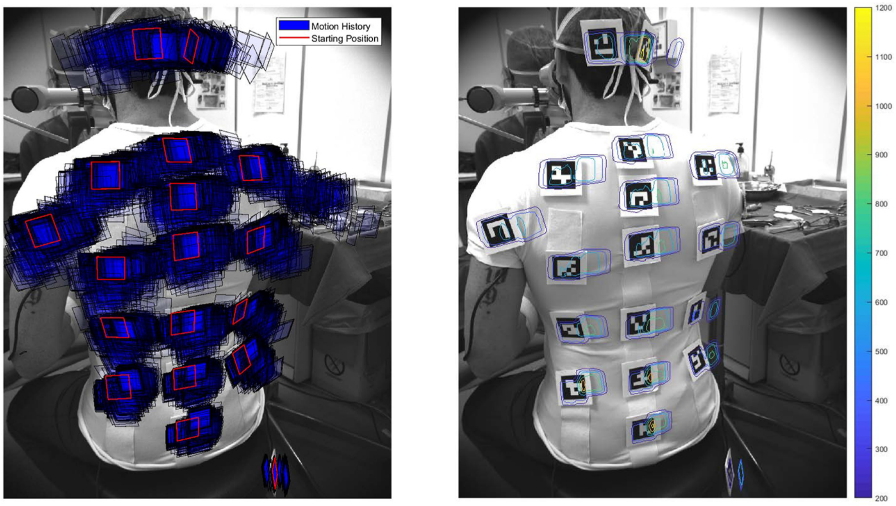


Figure 5.3: left: MHI in a left side loupes-aided operation; right: corresponding colormap for number of overlapping markers position

Loupes

In figure 5.3 a motion history example for a loupes-aided left side operation is shown. It can be seen that both neck and trunk are definitely less static than what they resulted in microscope-aided operations. However, the figure also shows that the starting reference position is still often represented and, therefore, assumed during the entire operation. This aspect is more visible analyzing the sample right side operation frame displayed in figure 5.4.

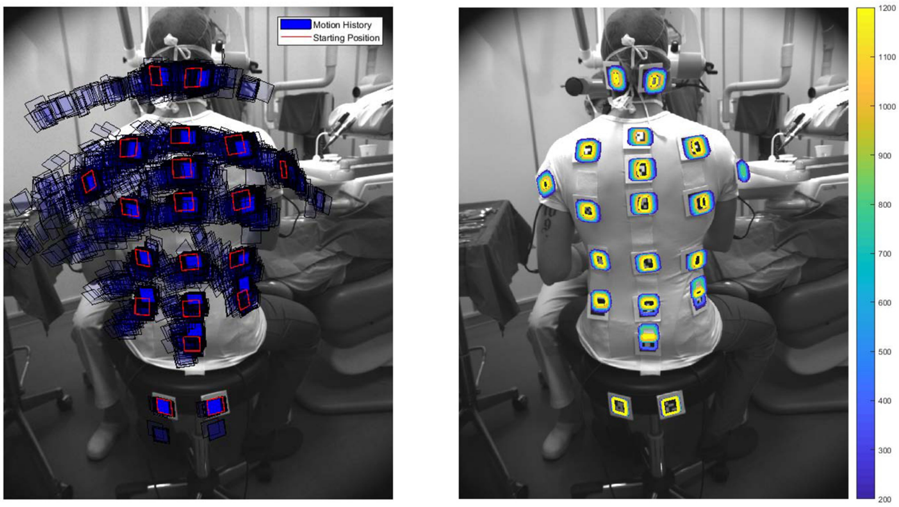


Figure 5.4: left: MHI in a right side loupes-aided operation; right: corresponding colormap for number of overlapping markers position

Naked eye

In figure 5.5 a motion history example for a left side operation performed without any magnification tool is shown. What can be assessed by looking at this figure is that the head and the trunk are definitely less static than they resulted with the other two methodologies. Despite it is difficult to read how close are the markers positions with respect to the reference position red squares, it is important to highlight that in this MHI intense blue areas are quite infrequent if compared with the all the other MHIs pictures considered so far. This means that marker positions overlap less frequently, therefore, when operating with naked eye, the dentist need to vary more often his body posture. Figure 5.5 displays a naked-eye right side operation: the obtained MHI reconstruction is coherent to what we saw for the left side operations. However, in the right side operation, a slightly higher level of repetition of the same marker positions is detectable for the bottom back markers.

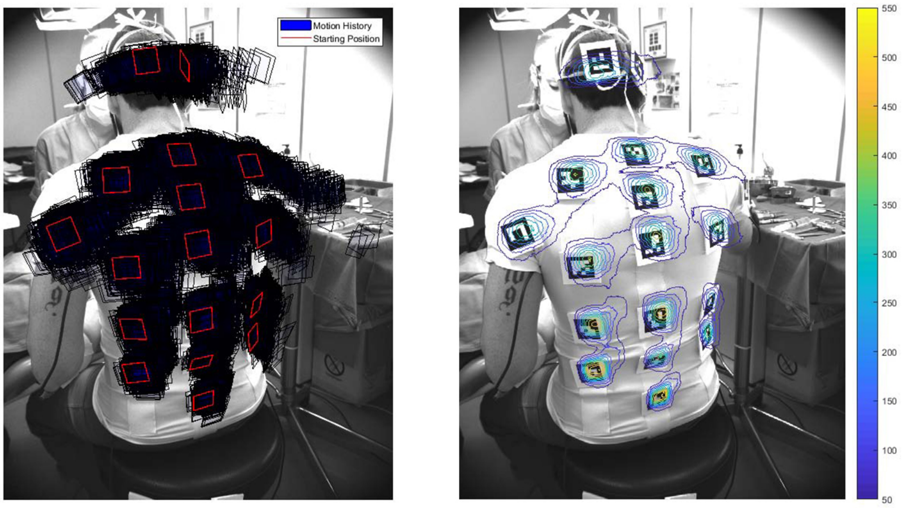


Figure 5.5: left: MHI in a left side operation performed with naked eye; right: corresponding colormap for number of overlapping markers position

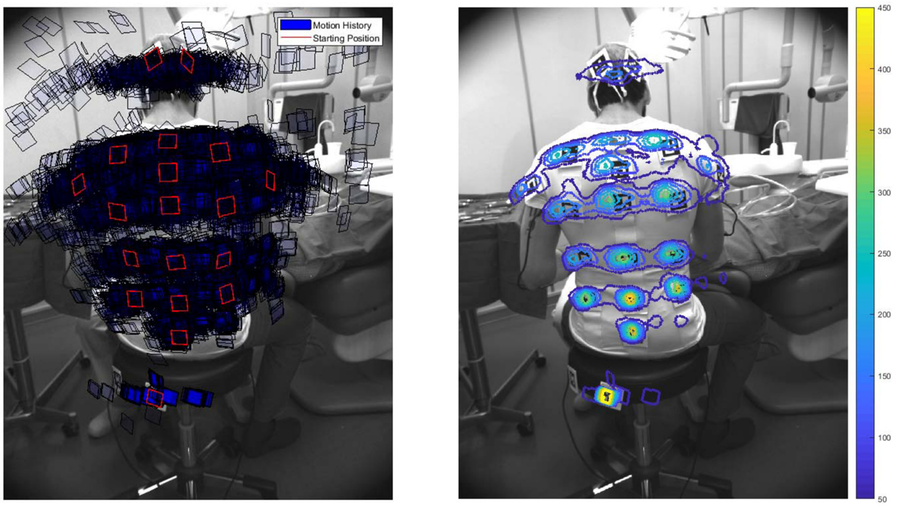


Figure 5.6: left: MHI in a right side operation performed with naked-eye; right: corresponding colormap for number of overlapping markers position

5.2 Angular comparisons

In this section we present the values of the RULA-relevant angles obtained from the analysis of the frames. In order to get a better representation and comparison, we decided to group the data by side of the operation and adopted visual aid. First of all we present the angle values with respect to the computed ground position, then we compare these values with their relative reference position.

The steps necessary for computing these angles have been introduced in previous chapter. The procedure presented eventually computed the 3D skeleton reconstruction image but in this chapter we focus onto the numerical results, useful for evaluating RULA scores.

5.2.1 Angle variations with respect to reference position

Microscope

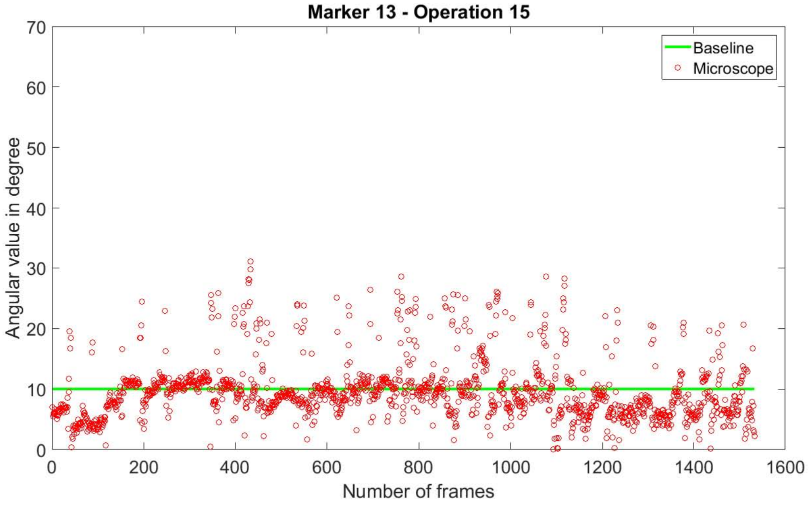


Figure 5.7: Neck inclination values during a left side microscope-aided operation, compared with the reference posture angle (green line)

In figure 5.7 the neck's inclination angles assumed during an entire left side microscope-aided operation are represented and the reference position's neck inclination value is also represented as a green line. Data indicate that the dentist kept a quite static neck posture during the operation as most of detected markers positions are grouped and close to the reference position angle. Some outliers are also present: they could be due to noise caused by some aleatory effects, such as a consistent head displacement from the microscope necessary to grab clinical instruments or they could be due to a wrong analysis of the algorithm caused by some light flashing on the marker.

In figure 5.8 there's a synthetic graph showing the trunk's inclination, side-bending and twist angles during the same operation, compared with the associated reference posture angles. What can be stated by looking at these data is that in this kind of operations side-bending is more frequent and consistent than twisting and we can expect twisting to often not give any contributions to the final RULA score associated with microscope operations.

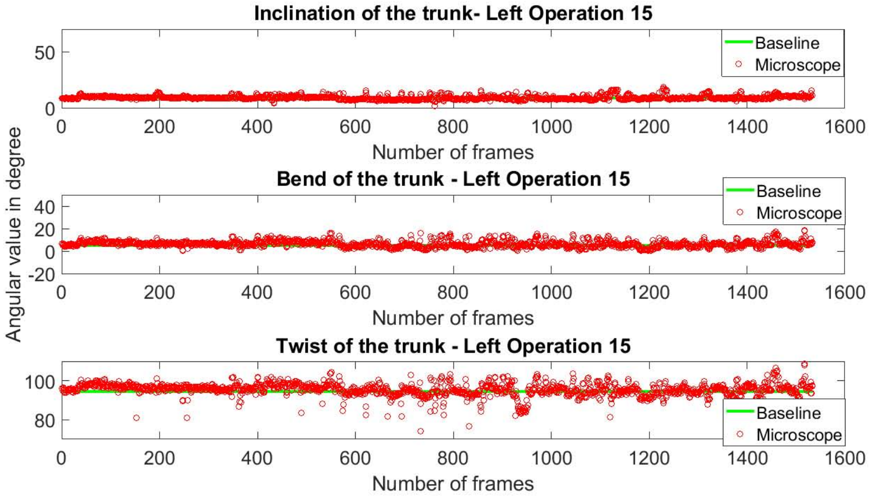


Figure 5.8: Trunk's inclination, side-bending and twist angles values during a microscope-aided operation, compared with the reference posture angles (green lines)

Figure 5.9 displays the neck inclination values registered during a right side microscope-aided operation. The graph confirms what seen already in figure 5.7 and the limited dynamism of the neck with this operation method.

Loupes

In figure 5.10 the neck's inclination angles assumed during an entire left side loupes-aided operation are represented and the reference position's neck inclination value is also represented as a green line. Data indicate a more dynamic behavior of the operator with respect to the previous methodology. In fact, data points appear to be less close to the reference position and less grouped, which indicates a more frequent and wide variations with respect to the reference inclination.

In figure 5.11 there's a synthetic graph showing the trunk's inclination, side-bending and twist angles during the same operation, compared with the associated reference posture angles. A main difference with

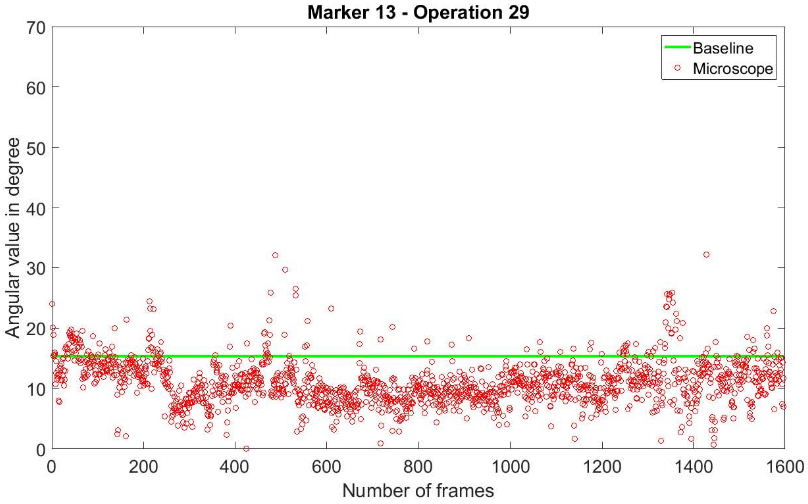


Figure 5.9: Neck inclination values during a right side microscope-aided operation, compared with the reference posture angle (green line)

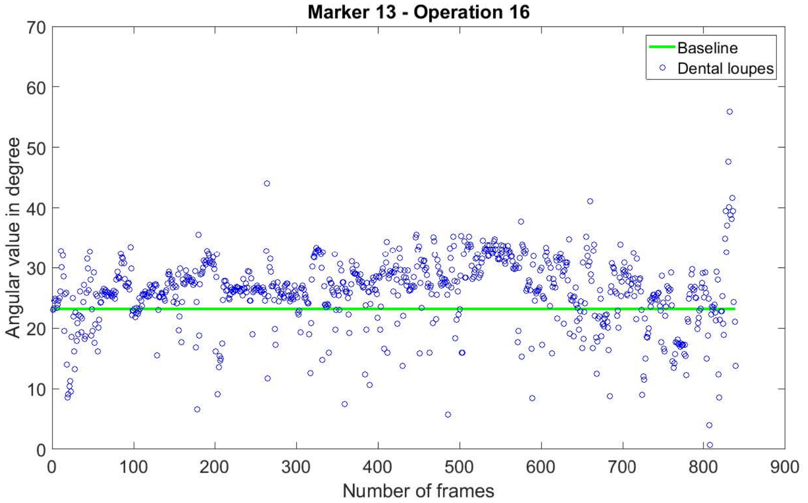


Figure 5.10: Neck inclination values during a left side loupes-aided operation, compared with the reference posture angle (green line)

respect to the microscope-aided operations is that of a more relevant presences of twisting angle and, in some cases, of side-bending angle.s

Naked eye

In figure 5.13 there is a graph of the data collected in a left-side operation performed with naked eyes, regarding the neck inclination angle. Data suggest that the operator had a quite dynamic behavior during the operation in comparison to the other two methodologies examined in our study. In fact, data points appear quite scattered and we cannot out-

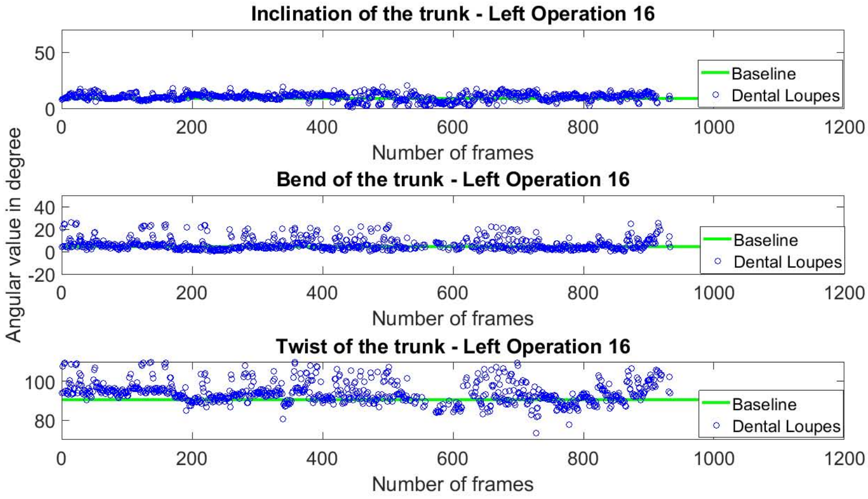


Figure 5.11: Trunk's inclination, side-bending and twist angles values during a loupes-aided operation, compared with the reference posture angles (green lines)

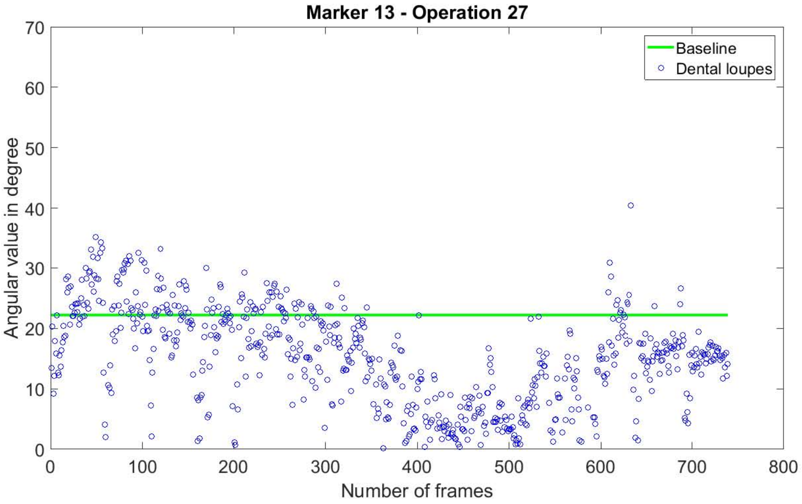


Figure 5.12: Neck inclination values during a right side loupes-aided operation, compared with the reference posture angle (green line)

line a clear trend in their distribution. This means that the variation of neck posture is very frequent and that inclination values wider that those encountered in the other approaches are also more frequent.

Analyzing the bend and twist of the trunk of figure 5.14, we see that also in this case twisting angle and side-bending angle appear having more relevant difference values with respect to the reference position. However, it has to be considered that these values take also into account the wider and unfrequent movements necessary for catching some

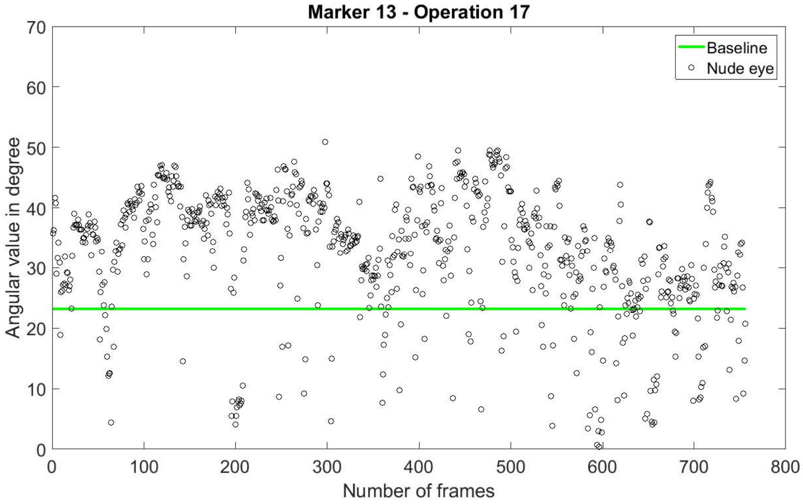


Figure 5.13: Neck inclination values compared with the reference posture angle (green line). Data are collected during a left side operation performed without external visual aids, just naked eyes.

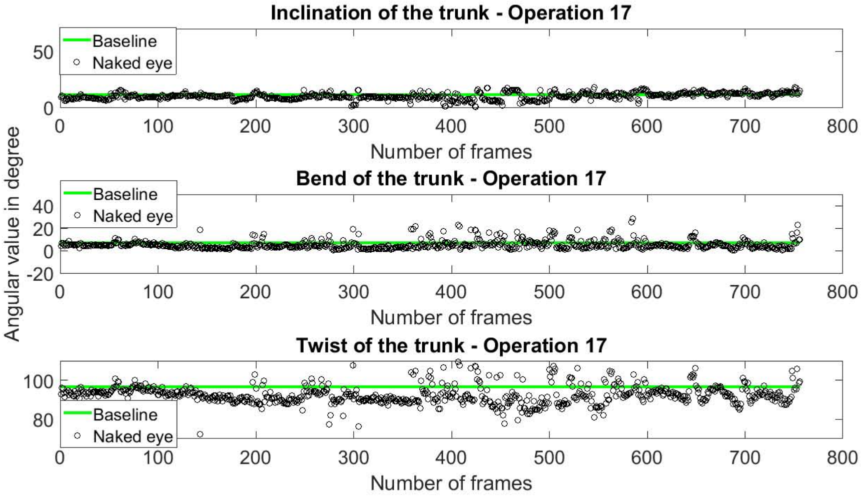


Figure 5.14: Trunk's inclination, side-bending and twist angles values during a naked-eye operation, compared with the reference posture angles (green lines)

instruments.

Finally, figure 5.15 displays the neck inclination values recorded for a right side operation performed with naked eyes which confirm the behavior seen for the left side case.

Comparison

In figures 5.16 and 5.17 there is a recapitalory comparison between the neck inclination values obtained respectively in a left side and in a right

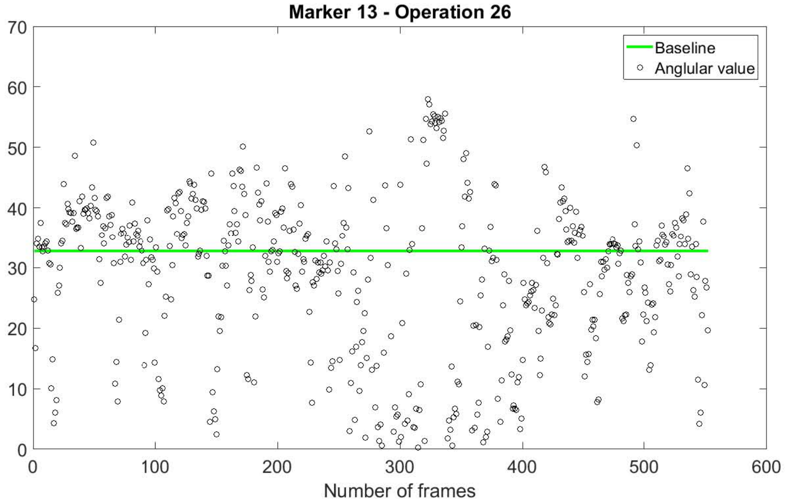


Figure 5.15: Neck inclination values compared with the reference posture angle (green line). Data are collected during a right side operation performed without external visual aids, just naked eyes.

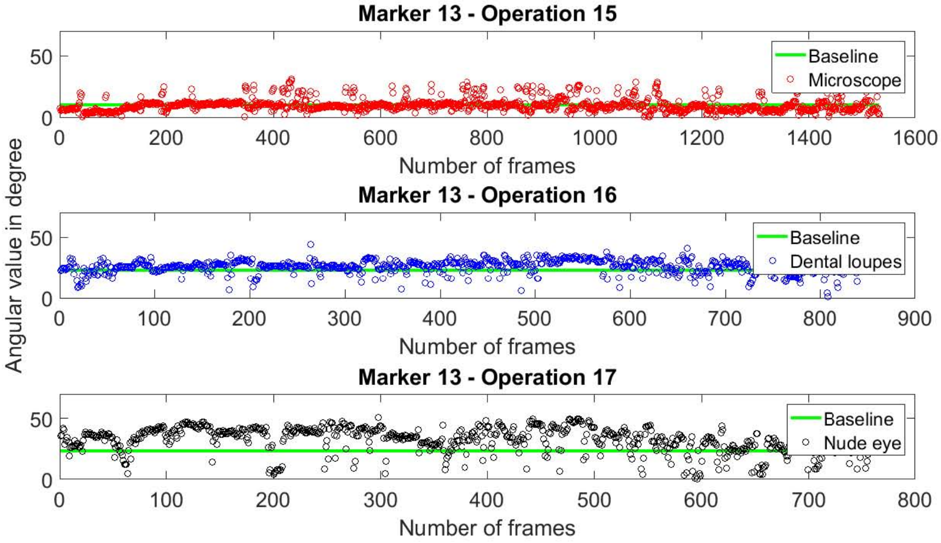


Figure 5.16: Comparison between neck inclination angle values recorded with the three different approaches for left side operations

side operation, using the three different approaches examined. These operations are the the same left operations considered in figure 5.18 and 5.18 where the probability density function of the neck angle values is represented. The analyses suggest that the microscope offers the best solution in terms of low dynamism and angles extension.

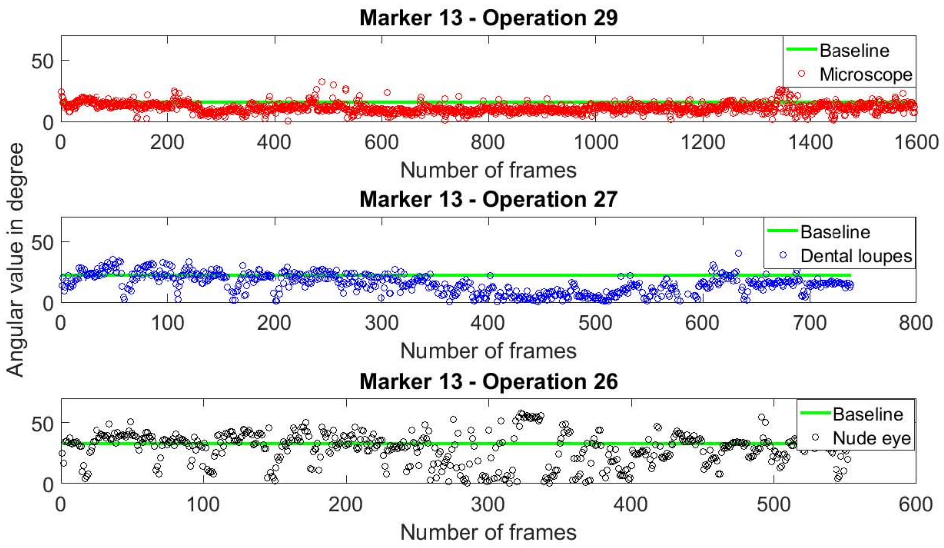


Figure 5.17: Comparison between neck inclination angle values recorded with the three different approaches for right side operations

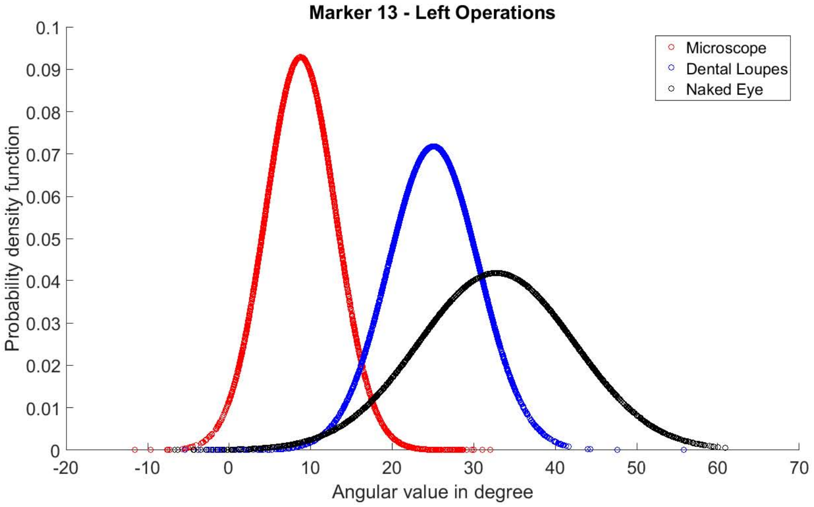


Figure 5.18: Probability density function of the distribution of the neck inclination angle values recorded with the three different method for left side operations

5.2.2 Angular distribution

The distribution of the angles is another important aspect to consider for our evaluation. In figure 5.18 there is a comparison between the distributions of the neck inclination angles recorded with the three different methods, regarding left side operations executed on the same day, so with exactly the same cameras settings.

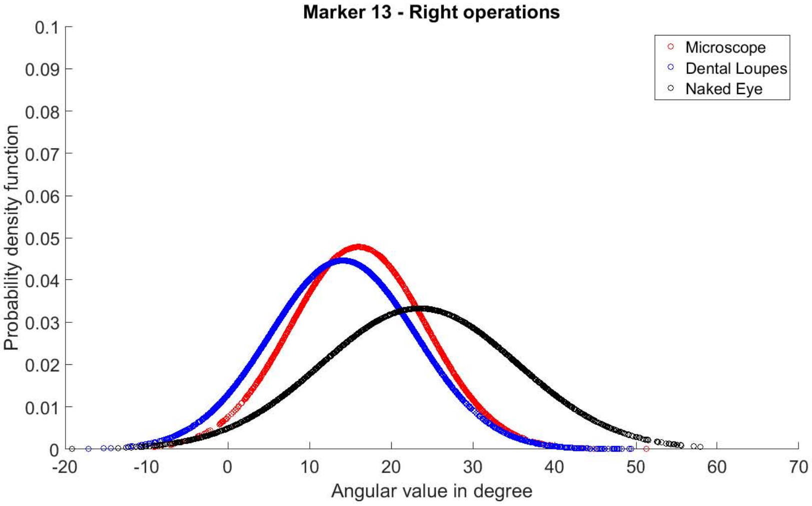


Figure 5.19: Probability density function of the distribution of the neck inclination angle values recorded with the three different method for right side operations

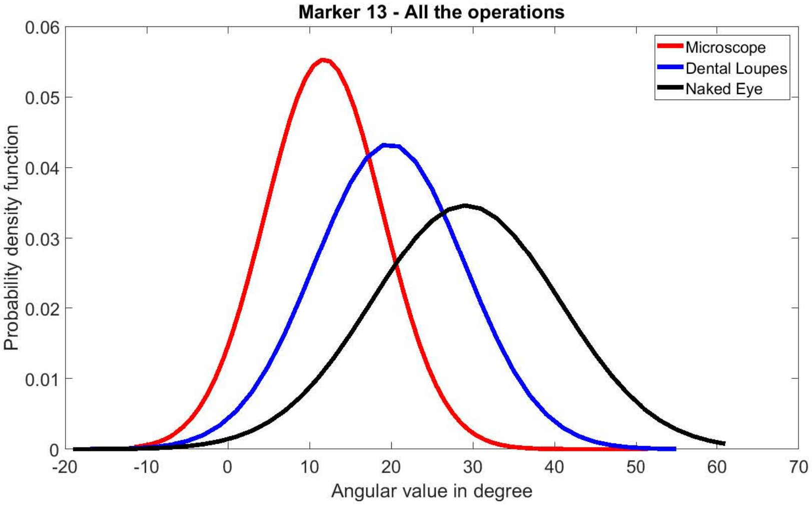


Figure 5.20: Probability density function of the distribution of the neck inclination angle values recorded in all the operations, grouped by methodology

By analyzing these results, we see that the curve relative to microscope-aided operations suggests us that the operator during this operation adopts a posture that allows him to keep the neck in a quite static position with an average inclination angle of around 9 degrees. This is indicated by the fact that the red Gaussian has a more tight bell curve and we have an higher probability to have the neck with that inclination.

If we consider the data regarding the operation executed using the dental loupes, we see that it has also a quite static posture as their bell curve is not as wide as the naked eye curve, despite being slightly wider that the microscope one. Besides, it is important to point out the notable difference in the average neck inclination values:in particular we have that the average value for loupes operation is about 27 degrees and that for naked eyes operations is about 35 degrees.

A comparison between the neck inclination angles values distribution for the three methods applied to right side operation is given in figure 5.19. In this case we see that the smallest average neck inclination angle is associated with loupes methodology, while that for microscope methodology is a few degrees higher: respectively, 13 and 18 degrees. These values describe an unusual but realistic situation that could be encountered sometimes but which is not reflected in most of the cases. For what concerns the width of the bell curves, also in this case the naked-eye curve is the widest one, confirming a higher dynamism required for this approach. These analyses, collected for all the operations, lead us to the global probability distribution of the angular values inclinations, shown in 5.20. We can see that the shapes of the bell curves are coherent with the results obtained with the sample operations considered so far. The global trend of the neck inclination angle is that the operator, when using the microscope, moves the neck less with respect to the other two techniques and keeps an less wide average inclination angle. In particular the average neck inclination angle values obtained by the total probability density functions are about 12 degrees for microscope, about 22 degrees for the loupes and 32 degrees for naked eyes.

5.3 RULA evaluation

In this section we evaluate the three different methods examined by following RULA criteria. In particular, since we focused our analysis on neck and trunk postures, we evaluate the results through step 9 and 10 of RULA algorithm.

The scores obtained are the following:

- Microscope: the neck score is between 1 and 2 while the trunk score is 1. By considering the values obtained for twist and side-bending angles for the trunk, we also assigned a score of 1 for their combined contribution;

- Loupes: the neck score is between 2 and 3 while the trunk score is in the range 2-4. Despite the twist and side-bending values for the trunk are slightly higher in this case, the score assigned is again 1 for their combined contribution, as their contribution is not strong enough for a higher score (RULA does not apply half point scores);

- Naked eye: the neck score is 3 and the trunk score is in the range 3-5. The score assigned for twist and side-bending of the trunk is 2, as they both often present a strong contribution in the frames position.

Analyzing the shape of the bell of the angular distribution presented in 5.20, we can also give the score accounted in RULA's step 6 which regards general muscle use: the score for microscope operations is 1 due to the fact that the dentist is quite static pfor most of the operations duration; the score assigned for loupes and naked eye methods is instead 0, despite loupes reveal to be a generally more static approach.

6
Conclusions

The purpose of this thesis project was that of realizing a reliable posture detection methodology which could allow a basic 3D body reconstruction meant to assess the quality of the posture assumed by the operating dentist during operations with the three visual solutions examined: naked eye, dental loupes and surgical microscope. The implemented procedure had to be non-invasive in order to not disturb the operator and it was applied through the combined use of a markers-equipped t-shirt (to be worn by the dentist) and two cameras.

We collected the 3D markers position parameters by using this equipment together with a pc and our C++ script based on OpenCV libraries. Then we sorted them into the three visual aids categories, after separating those related with right side operations from those related with left side operations, which, due to the operating room instruments arrangement, required two slightly different setups. With the data collected and sorted we could proceed to the analysis phase in which the quality of the postures assumed throughout each operation were evaluated with the aid of graphical solutions as the motion history images which already allowed us a qualitative evaluation of the average extension of the movements of the operator and of his level of dynamism. The other graphical tool used for analyzing the data was that of 3D skeleton reconstruction which were the key step necessary for computing the posture assessment scores contemplated by RULA.

From a comparison of the processed data and the RULA scores obtained by the three different methodologies compared in our study, it clearly emerged that the visual aid used in the operations by the dentist strongly affected his movements and the quality of his posture. In particular we found the following results for the different visual aids:

- The microscope data analyses suggest that the dentist assumes an average neck position between 10 and 15 degrees for most of the operation, the whole motion is quite static and it lies in a small position interval around the reference position;

- The dental loupes data analyses show that the dentist assumes an average neck position between 20 and 25 degrees for most of the operation, the general motion is less static, it introduces more twisting and side-bending factors contributions but it still lies in a small (despite larger than microscope one's) position interval around the reference position;

- The naked eye data analyses show that the dentist assumes an average neck position between 23 and 30 degrees for most of the operation, the general motion is quite dynamic and it concerns a large position interval around the reference position. Besides, both twist and side-bending factors are increased with respect to the other two methodologies results;

The results obtained by assessing our data with RULA algorithm also confirmed the better quality of the general posture assumed during an operation performed with the microscope with respect to loupes and naked eyes operations. Considering the RULA scores obtained considering only the upper body evaluation, we could determine that the surgical microscope is the best visual aid in terms of quality of the posture assumed during the operations. In fact it is the methodology that, according to RULA, achieves the lowest levele of musculoskeletal disorders risks. These results are coherent with the starting trend forecasts and with the general feeling of the doctor. Future developments of this work could involve developing a system for monitoring the movement of the legs,arms and wrists during the operations, in order to cover the remaining scores of the RULA. A possible implementation solution could be through the use of accelerometers in combination with the ArUco makers. In this way we could achieve a more accurate posture reconstruction which is the key element for assessing the quality of human body posture in order to evaluate and avoid the risk of MSDs. The procedures described and implemented in these project and their possible future improvements could be usefully applied in many areas of interest of ergonomics, providing an effective tool for investigation of MSD risks in several job areas.

List of Figures

Bibliography

[1] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[2] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.

[3] D. A. Forsyth and J. Ponce, “A modern approach,” *Computer vision: a modern approach*, vol. 17, pp. 21–48, 2003.

[4] J. N. Cederberg, *A course in modern geometries.* Springer Science & Business Media, 2013.

[5] S. Belongie, “Rodrigues’ rotation formula,” *From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld. wolfram. com/RodriguesRotationFormula. html*, 1999.

[6] A. De la Escalera and J. M. Armingol, “Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration,” *Sensors*, vol. 10, no. 3, pp. 2027–2044, 2010.

[7] J. Canny, “A computational approach to edge detection,” in *Readings in computer vision*, pp. 184–203, Elsevier, 1987.

[8] R. Deriche, “Using canny’s criteria to derive a recursively implemented optimal edge detector,” *International journal of computer vision*, vol. 1, no. 2, pp. 167–187, 1987.

[9] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[10] J. Wang, F. Shi, J. Zhang, and Y. Liu, “A new calibration model of camera lens distortion,” *Pattern Recognition*, vol. 41, no. 2, pp. 607–615, 2008.

[11] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment—a modern synthesis,” in *International workshop on vision algorithms*, pp. 298–372, Springer, 1999.

[12] K. I. McKinnon, “Convergence of the nelder–mead simplex method to a nonstationary point,” *SIAM Journal on optimization*, vol. 9, no. 1, pp. 148–158, 1998.

[13] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[14] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, “Speeded up detection of squared fiducial markers,” *Image and Vision Computing*, vol. 76, pp. 38–47, 2018.

[15] J. Köhler, A. Pagani, and D. Stricker, “Detection and identification techniques for markers used in computer vision,” in *Visualization of Large and Unstructured Data Sets-Applications in Geospatial Planning, Modeling and Engineering (IRTG 1131 Workshop)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2011.

[16] M. Fiala, “Artag, a fiducial marker system using digital techniques,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 2, pp. 590–596, IEEE, 2005.

[17] H. Kato and M. Billinghurst, “Marker tracking and hmd calibration for a video-based augmented reality conferencing system,” in *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR’99)*, pp. 85–94, IEEE, 1999.

[18] C. Li and S. Lee, “Computer vision techniques for worker motion analysis to reduce musculoskeletal disorders in construction,” in *Computing in Civil Engineering (2011)*, pp. 380–387, 2011.

[19] P. Van Wely, "Design and disease," *Applied Ergonomics*, vol. 1, no. 5, pp. 262–269, 1970.

[20] J. Perrott, "Anatomical factors in occupational trauma," *Medical journal of Australia*, vol. 1, no. 3, pp. 73–82, 1961.

[21] M. Melchior, Y. Roquelaure, B. Evanoff, J.-F. Chastang, C. Ha, E. Imbernon, M. Goldberg, and A. Leclerc, "Why are manual workers at high risk of upper limb disorders? the role of physical work factors in a random sample of workers in france (the pays de la loire study)," *Occupational and environmental medicine*, vol. 63, no. 11, pp. 754–761, 2006.

[22] B. Valachi, "Ergonomics and injury in the dental office," *RDH*, vol. 28, no. 4, p. 27, 2008.

[23] G. Li and P. Buckle, "Current techniques for assessing physical exposure to work-related musculoskeletal risks, with emphasis on posture-based methods," *Ergonomics*, vol. 42, no. 5, pp. 674–695, 1999.

[24] L. McAtamney and E. N. Corlett, "Rula: a survey method for the investigation of work-related upper limb disorders," *Applied ergonomics*, vol. 24, no. 2, pp. 91–99, 1993.

[25] A. Sanchez-Lite, M. Garcia, R. Domingo, and M. A. Sebastian, "Novel ergonomic postural assessment method (nerpa) using product-process computer aided engineering for ergonomic workplace design," *PloS one*, vol. 8, no. 8, p. e72703, 2013.

[26] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.", 2008.

[27] H. J. Vala and A. Baxi, "A review on otsu image segmentation algorithm," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 2, pp. 387–389, 2013.

[28] O. Rodrigues, *Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace: et de la variation des cordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire.* 1840.

[29] M. A. R. Ahad, J. K. Tan, H. Kim, and S. Ishikawa, “Motion history image: its variants and applications,” *Machine Vision and Applications*, vol. 23, no. 2, pp. 255–281, 2012.

[30] G. R. Bradski and J. W. Davis, “Motion segmentation and pose recognition with motion history gradients,” *Machine Vision and Applications*, vol. 13, no. 3, pp. 174–184, 2002.