Camera Calibration & Structure from Motion Procedures Using Humanoid NAO

Ozan Gönenç, Prof. David Fofi and Dr. Ralph Seulin
Laboratoire Le2i UMR 5158 CNRS, Université de Bourgogne

Abstract— **This paper presents the applications of Camera Calibration and Structure from Motion(SfM) procedures for Humanoid Nao. The camera calibration of mobile robots are mostly done using manual approaches. Unlike traditional camera calibration applications for humanoids, our application provides a fully automatic camera calibration module for Nao that provides reliable and accurate results. The second application we present is an optimal method for SfM which uses the results of the calibration process and which is applicable to a humanoid. A prototype of the application of SfM for NAO is presented. We also present real-word experiments with our humanoid in order to evaluate our approach and results.**

I. INTRODUCTION

The aim of humanoid robotics research is to gain better understanding of human body structure and behaviours and create a humanoid that reaches the capabilities of a human in terms of motion, perception, information interpretation and intelligence. Nao is a 57 cm tall humanoid, with 27 degrees of freedom, produced by Aldebaran Robotics[**?**] and it is one of the best of its league.

For most of the computer vision applications, camera calibration is the first step because camera parameters are required to measure the metric structure of a scene precisely. Camera calibration is the process of obtaining camera intrinsic and extrinsic parameters and camera parameters are necessary to obtain knowledge about the euclidean structure of the 3D world from 2D images. Thus to be able to obtain metric information from images, one needs to know the camera parameters. The first part of the paper presents our application for the camera calibration algorithm. The algorithm is developed by Z. Zhang[28] and our application is based on this method.

Structure from Motion(SfM) is the method of reconstructing 3D points and recovering the camera motion from 2D images that are taken from different position of the same scene. The reconstruction of rigid objects and motion estimation of the camera, from a sequence of images is a challenging and a popular problem. We present an algorithm for SfM that is an optimal approach for an application on Nao.

II. CAMERA CALIBRATION

There exists different techniques for camera calibration and these techniques might be categorized by different criterion. Most general classification of those algorithms would be photometric calibration and self-calibration. Photometric calibration refers to the calibration algorithms that are using some apparatus for camera calibration. These apparatus are 3D objects with expensive setups'and a very precise metric information of those setups'needs to be known for those calibration methods. Tsai[26] proposed a method that uses a 2D pattern but the metric information about the motion of the pattern should be known precisely. Thus there is no difference between using a 3D object. The method assumes that some of the camera parameters are provided by the manufacturer in order to reduce the initial guess of parameters. It is based on a pinhole perspective projection model and the parameters to be estimated are focal length of the camera, radial distortion coefficient, rotation and translation components and coordinates of centre of radial lens distortion. The technique uses a 2 step approach. First step includes the computation of all the extrinsic parameters except t_z. This process is a solution of a system of linear equations. The input of these equations are the coordinates of the points both in the calibration pattern and world coordinate systems. The second step is to estimate all the missing parameters by non-linear optimization using Levenberg-Marquadt algorithm[18]. Faugeras[4] uses two orthogonal planes with A 3D coordinate system with known coordinates and known the corners in this coordinate system. For more extensive and detailed comparisons and evaluations of the existing camera calibration techniques see [23] and [17].

Self-calibration is the process of obtaining the camera parameters using sequences of images from the camera without any known structure or using any specific apparatus. Providing on-the-fly camera calibration, self-calibration techniques([5], [9]) are useful but still there are robustness issues of existing self calibration methods. Most of the time these issues are related to the geometry of the scene image contents that will be used for recovering the camera calibration parameters. That's why, camera self-calibration methods are not used.

For our application we use the method developed by Z. Zhang[28] because this method do not require any setup and only needs a planar pattern to be observed from different angles. For this reasons this method is the most suitable one for Nao.

A. Z. Zhang's Camera Calibration Method

The method, developed by Z.Zhang[28] is an easy to implement technique that also requires an easy setup; only a planar pattern as an apparatus to be observed from different positions of the camera. The pattern could be anything with the condition of known metric on the plane but traditionally the chessboard pattern is used because it is easy to extract information from the chessboard pattern.

1) Notation: 2D points are denoted by lower letters and 3D points are denoted by capital letters. The homogeneous coordinates of a 2D point is denoted by x̃ and a 3D point is denoted by X̃. Thus $m = [u, v]^T$ denotes a 2D point and m̃= $[u, v, 1]^T$ denotes the same 2D point with homogeneous coordinates. The notation is same for 3D points as well. A camera is modelled by the usual pinhole model. The relationship between a 3D point M and corresponding image projection m is given by

$$s\tilde{m} = A\begin{pmatrix}R & t\end{pmatrix}\tilde{M} \tag{1}$$

$$A = \begin{pmatrix}\alpha & \gamma & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{pmatrix}$$

where (R, t) of equation 1 is the extrinsic parameters of the camera and s is an arbitrary scale factor. A is the camera intrinsic parameters matrix. α and β of A are the scale factors in u and v direction respectively and u_0 and v_0 are the coordinates of the principal point. Note that $(A^{-1})^T$ or $(A^T)^{-1}$ is denoted by A^{-T}.

2) Methodology: Firstly, it is assumed that the model plane is placed on $Z = 0$ of the world coordinate system. Hence, when we denote the columns of rotation matrix R separately from equation 1 we get

$$s\begin{pmatrix}u\\ v\\ 1\end{pmatrix} = A\begin{pmatrix}r_1 & r_2 & r_3 & t\end{pmatrix}\begin{pmatrix}X\\ Y\\ 0\\ 1\end{pmatrix}$$

$$= \underbrace{A\begin{pmatrix}r_1 & r_2 & t\end{pmatrix}}_{\text{H}}\begin{pmatrix}X\\ Y\\ 1\end{pmatrix}$$

From above equations the homography H we obtain is

$$H = A\begin{pmatrix}r_1 & r_2 & t\end{pmatrix} \text{ thus } s\tilde{m} = H\tilde{M} \tag{2}$$

An homography can be estimated from the image of the model plane. The method utilizes the technique for estimation of homography based on maximum likelihood criterion. Thus from 2

$$\begin{pmatrix}h_1 & h_2 & h_3\end{pmatrix} = \lambda A\begin{pmatrix}r_1 & r_2 & t\end{pmatrix}$$

where λ is an arbitrary scalar because H is defined up to a scale factor. Since r_1 and r_2 are orthonormal, one can define two constraints on the intrinsic parameters.

$$h_1^T A^{-T} A^{-1} h_2 = 0 \tag{3}$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \tag{4}$$

Notice that for both constraints that $A^{-T}A^{-1}$ is the image of absolute conic[8]

3) Solving Camera Calibration: This section provides the explanation of the closed-form solution of the camera calibration problem[28]. The analytical solution and the non-linear optimization technique will be explained.

The constraints of the intrinsic parameters include the image of absolute conic($A^{-T}A^{-1}$). Thus let

$$B = A^{-T}A^{-1} \equiv \begin{bmatrix}B_{11} & B_{12} & B_{13}\\ B_{12} & B_{22} & B_{23}\\ B_{13} & B_{23} & B_{33}\end{bmatrix} \tag{5}$$

Now notice that B is a 3×3 symmetric matrix with 6 independent entries. Then B can be defined as a 6D vector with the upper triangular elements of B

$$b = \begin{bmatrix}B_{11} & B_{12} & B_{22} & B_{12} & B_{23} & B_{33}\end{bmatrix}^T \tag{6}$$

Then we have

$$h_i^T B h_j = v_{ij}^T b \tag{7}$$

with

$$v_{ij} = \begin{bmatrix}h_{i1}h_{j1}\\ h_{i1}h_{j2} + h_{i2}h_{j1}\\ h_{i2}h_{j2}\\ h_{i3}h_{j1} + h_{i1}h_{j3}\\ h_{i3}h_{j2} + h_{i2}h_{j3}\\ h_{i3}h_{j3}\end{bmatrix}^T$$

where h_i denotes the i^{th} column vector of H. Now we can when we rewrite the two constraints 3 and 4 as

$$\begin{bmatrix}v_{12}^T\\ (v_{11} - v_{22})^T\end{bmatrix} b = 0 \tag{8}$$

Thus we have 2 equations for each image of the model plane. When we have n images we will stack all the equations and will obtain $2n \times 6$ matrix V as shown in equation 9.

$$Vb = 0 \tag{9}$$

V is a $2n \times 6$ matrix. Applying SVD we get the solution to 9. Knowing b, and $B = \lambda A^{-T}A$ we can obtain all the intrinsic parameters. Finally the extrinsic parameters for each image can be computed from A. From 2

$$r_1 = \lambda A^{-1}h_1$$
$$r_2 = \lambda A^{-1}h_2$$
$$r_3 = r_1 \times r_2$$
$$t = \lambda A^{-1}h_3$$

where $\lambda = 1/||A^{-1}h_1|| = 1/||A^{-1}h_1||$

B. Camera Calibration Procedure using Nao

This part explains a two-phased application method for the camera calibration to Nao. The first phase is the acquisition of chessboard pattern images from different positions of the camera and the second phase is camera calibration using the images acquired in the first phase.

In the first phase there are three main considerations and these are corner detection for correct image acquisition, motion and image acquisition. The procedure in this phase is repeated for a given number of times in order to obtain required number of images.

- **Chessboard Corner Detection**
  One has to be sure that the planar chessboard pattern is in front of the NAO's camera perspective and the corners of the pattern are extractable. Both of the goals are achieved by employing a corner detection method since when there is no pattern in front of the camera, there will not be any corners to detect. If we do not make sure that the corners are extractable then we may acquire irrelevant sequence of images that has no value for the calibration process. The corner detection method is the findChessBoardCorners method of OpenCV. The method first applies a thresholding, then erosion. The aim for both processes is to break the connectivity between the squares. Then the method tries to obtain the input numbers for number of squares. The erosion process is repeated as many times as it requires to break the connectivity and obtain the required number of squares. Finally, the squares are brought back together to form the board pattern.
- **Motion**
  We need to move Nao in order to obtain an image from different positions. Two kinds of motion is programmed. The first motion is only body motion[1]. This motion is based on the knees and the pelvis. The second motion is walking in a circular fashion(This motion is illustrated in Figure 1). The motion of Nao is programmed using the built-in functions of Nao and any motion that contains rotation and translation at the same time is valid.

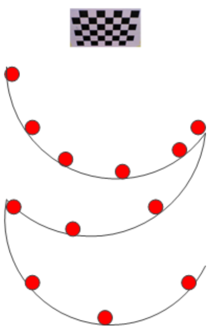

Fig. 1. The illustration of motion of Nao in order to take different positions for image acquisition. The red points represent Nao.

[1]Without moving Nao by the usege of it's feet

- **Image Acquisition**
  After a particular motion, if the corners of the chessboard pattern is extractable, we will acquire the image and save it. OpenCV and Python Image Library is used for image acquisition and saving.

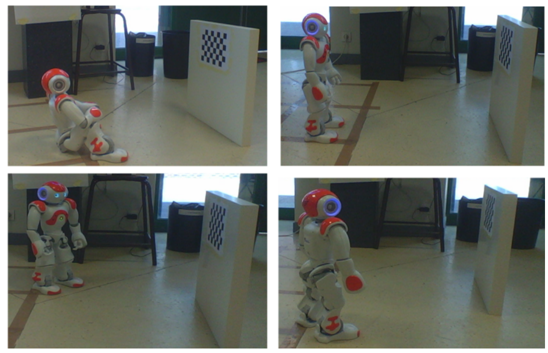

Fig. 2. 4 different Poses of NAO in order to get images

In the second phase of the calibration procedure we only need to read images that are saved, and employ the calibration procedure. Note that in order to obtain the pattern images there are two different options. One may either fix the pattern and move the camera or fix the camera and move the pattern. We have applied and tested both procedures but the goal of our calibration process in terms of motion is the former one. The technical details of the implementation of the method is explained in experiments and results section.

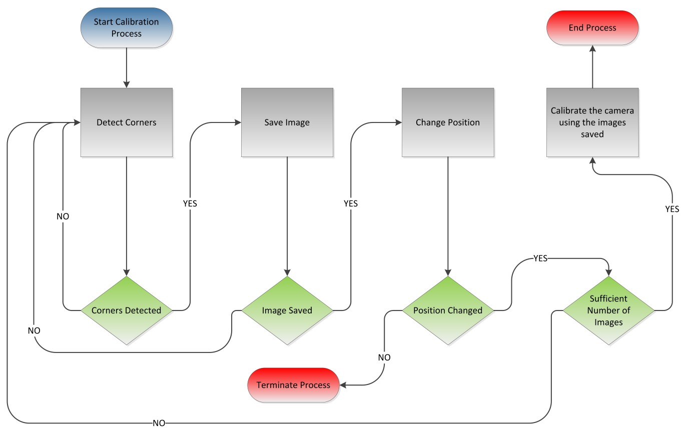

Fig. 3. The calibration procedure

III. Structure from Motion

Structure from Motion is the method of reconstructing 3D points and recovering the camera motion from 2D images that are taken from different position of the same scene. The reconstruction of rigid objects and motion estimation of the camera, from a sequence of images is a challenging and a popular problem in computer vision. There exists a variety of approaches after years of research on the topic. To classify SfM algorithms one can define two general conditions in terms of camera characteristics. *Calibrated SfM* where the camera matrix of the camera is known and *uncalibrated SfM*.

Calibrated SfM results with euclidean structure of the scene, thus using calibrated SfM it is possible to obtain real measurements such as lengths and angles of the real scene. The most popular calibrated SfM methods are the factorization methods([3],[11],[25] to cite a few). The common point of all factorization methods is that they all allow to linearize the camera observation model and their computational cost is low, thus they provide fast results. Another method is called generic SfM. The main property of the generic SfM is that those techniques can be applied to all kinds of cameras(pinhole cameras, catadioptric, omnidirectional etc.). A generic camera calibration and SfM technique is defined by Ramalingam *et al.*[22]. The problem of the method is the failure of the correspondence algorithm(SIFT). Another generic SfM method is defined by Mouragnon *et al*[19]. Their method is a generic real-time SfM based on incremental 3D reconstruction.

The uncalibrated SfM methods assume that the camera parameters are not known, thus using those methods one can only obtain projective reconstructions. It is impossible to obtain real measurements such as lengths and angles of the real scene. In order to obtain euclidean reconstruction, the camera parameters should be given. The most popular uncalibrated SfM is the factorization based methods, as it is for the calibrated SfM. A very popular method for factorization based uncalibrated SfM is proposed by Sturm and Triggs[24]. Note that for this method and initial guess of projective depth is necessary. In [16], in order to improve the results of [24] some iterative methods have proposed. Oliensis and Hartley [21] show that [24],[16] are unstable because of the convergence problem and in [21] they proposed a new iterative extension for [24] and [16]. There are more techniques based on bilinear programming, Direct metric structure and bundle adjustment, etc., and an extensive survey is done by [12] explaining each category of algorithm in more details.

A. The general method of SfM

A three-dimensional scene point can be reconstructed using image pairs. There are parameters that define type of reconstruction and these are; *the fundamental matrix* and *camera matrix*. Since this is a general method, it is assumed that there is no knowledge about the cameras or the scene and the method is explained.

Given an image pair of the same scene there exists correspondences $x_i \longleftrightarrow x_i^{'}$ where x_i and $x_i^{'}$ are the corresponding points in the images. Then for all i we have the following relationship between the corresponding points and the actual, unknown, 3D points X_i

$$x_i = PX_i \text{ and } x_i' = P'X_i \text{ for all i}$$

The method of 3-dimensional reconstruction can be summarized as;

1) *Compute the fundamental matrix from point correspondences*
   The fundamental matrix F satisfies $x_i' F x_i = 0$ when the correspondences are known, and each point correspondence one linear equation for F, thus by knowing at least 8 corresponding point it is possible to solve the equation linearly for F. Generally we will have more than 8 points thus, the least square solution is the singular vector corresponding the smallest singular value of A.

2) *Compute the camera matrices from fundamental matrix*
   The camera will be chosen as $P = \big[I|0\big]$ and $P' = \big[[e']_\times F|e'\big]$

3) *Compute the 3D point X_i which corresponds to the relation $x_i \longleftrightarrow x_i^{'}$*
   The process is also known as *triangulation* and the solution of triangulation is; since $x = PX$ and $x' = P'X$, one should combine these two equations into a form of $AX = 0$. Then solve $AX = 0$ using SVD and picking the singular vector corresponding to the smallest singular value.

B. An Algorithm for SfM on NAO

In the previous sections, 3D reconstruction problem is discussed and the SfM methods classified using certain criterions. Now, we will complete our discussion by explaining the details of our approach applied to Nao. Our method is a calibrated SfM method because the camera parameters are obtained by the method that we have explained in the previous section will be used. There are four steps of the method and these are corresponding point detection and matching, essential matrix estimation, camera extraction and triangulation

1) The Correspondence Problem: The correspondence problem is the problem of finding the corresponding points of two images that are acquired by a moving camera. After years of research on correspondence problem, there are a lot of different methods are developed and the most popular methods are those based on Harris'operator[7], SIFT[2][15] and SURF[3][1]

The method that we choose to find corresponding points in a sequence of images, is based on feature detection and feature descriptor creation. It is called speeded up robust features(SURF)[1]. Assume we have only two images and we want to find the corresponding points for these images. SURF first detects potential feature points in the first image and defines descriptors for each detected feature point. Then the same procedure is applied to the second image. The image descriptors are defined by dividing a local image into a grid(typically 4x4) and a orientation histogram is computed for each of these cells. Then, knowing each feature point's descriptor for both images, the best matches are found. Finally RANSAC[6] is used to filter positive false matches.

[2]Scale-invariant feature transform
[3]Speeded up robust features

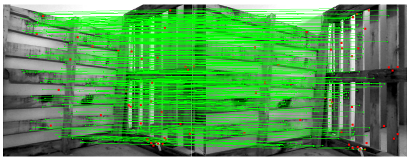

Fig. 4. Feature point detection and matching using SURF

Note that the computational cost of SURF descriptor is low. Since the methods are applied to a humanoid(mobile robot) SURF descriptors are chosen because on a humanoid, in motion, the time of the computation is crucial.

2) Essential Matrix Estimation: The fundamental matrix(F) is a 3×3 the algebraic representation of the epipolar geometry and the notion of F is defined for any corresponding points $x \longleftrightarrow x'$ in two images as

$$x'^T F x = 0 \tag{10}$$

then, the essential matrix constraint(a specialization of fundamental matrix constraint) is defined for calibrated cases as

$$\hat{x}_i'^T E \hat{x}_i = 0 \tag{11}$$

and the essential matrix equation can be written as

$$E \equiv [t]_\times R \tag{12}$$

Normally F has 8 degree of freedom(DoF) but in the special case of essential matrix where the points are defined in the camera coordinate system by applying the K^{-1}(the inverse of camera intrinsic parameters matrix) to the corresponding points x_i and x_i', the DoF of the essential matrix becomes 5, 3 DoF for translation and 3 DoF for rotation matrices(see equation 12), since the essential matrix is defined up to a scale. This means that given at least 5 corresponding points in two image, one can estimate the essential matrix. The algorithm, for estimating the essential matrix, by using at least 5 points is called the five-point algorithm[20]. However, we do not employ five-point algorithm it proposes a non-linear solution to the essential matrix estimation problem. A non-linear solution is not suitable for our purposes since computation cost of the non-linear estimation methods very high. The method is called by humanoid robot in motion, thus we need a faster and computationally inexpensive solution. Moreover, the non-linear estimation methods are sophisticated methods that are highly complex to implement.

Due to the reasons that we have explained the normalized eight-point algorithm[14],[10] is chosen to estimate the essential matrix. The normalized eight-point algorithm has advantages over five-point algorithm and it suits our purposes for the reasons such as; it solves the essential matrix estimation problem linearly consequently it is fast and computationally inexpensive and it is easy to implement.

3) Camera Extraction from Essential Matrix: The essential matrix is used to recover rotation(R) and translation(t) vectors.

> ***Theorem*** *Let the singular value decomposition of the essential matrix be* $E \sim U diag(1,1,0)V^T$, *where* U *and* V *are chosen such that* $det(U) > 0$ *and* $det(V) > 0$. *Then* $t \sim t_u \equiv \begin{bmatrix} u_{13} & u_{23} & u_{33} \end{bmatrix}^T$ *and R is equal to* $R_a \equiv UDV^T$ *or* $R_b \equiv UD^TV^T$

As explained in the theorem by David Nipster[20], any combination of the R and t satisfies the epipolar constraint. Thus there is an ambiguity and to find the true combination the first camera matrix is assumed to be $[I|0]$ and t is unit length. There are four possible solutions except for overall scale, and these are; $P_1 \equiv \begin{bmatrix} R_a & | & t_u \end{bmatrix}$, $P_2 \equiv \begin{bmatrix} R_a & | & -t_u \end{bmatrix}$, $P_3 \equiv \begin{bmatrix} R_b & | & t_u \end{bmatrix}$ and $P_4 \equiv \begin{bmatrix} R_b & | & -t_u \end{bmatrix}$.
The true configuration is decided using the cheirality constraint[4]. Note that one point that satisfies the cheirality constraint, is enough to decide which configuration is the right one. In order to test the cheirality the point is triangulated[8]. $([I|0], P_1)$ view pair is used first to find the 3D point Q. The points for the two views are indicated as Q_1 and Q_2. If $Q_1 Q_2 > 0$ and $(P_i Q_1) Q_2 > 0$ where i is one of the possible solutions, then the cheirality constraint is satisfied and the positions of the cameras are found. Finally, simple triangulation by back-projecting rays from the image points will fail because of the noise in the image points and the back-projected rays will not intersect. Thus triangulation requires definition and solution of a cost function. There are different solutions of triangulation. We have employed a linear triangulation method which is a direct analogue from DLT method[8]. Since we have the relations $x = PX$ and $x' = PX'$ for the first and second image points these equations can be combined as $AX = 0$ which is an equation linear in X. And as a result we obtain the 3D point cloud of the scene.

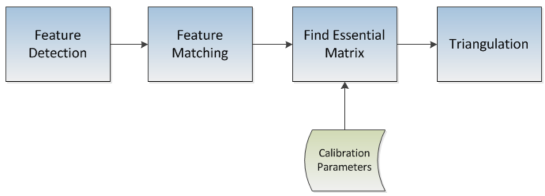

Fig. 5. The SfM algorithm

IV. EXPERIMENTS AND RESULTS

The calibration method is implemented using python[27] scripting language. Python was chosen as it is supported by Naoqi framework[**?**]. In addition to the built-in functions that Aldebaran Robotics provided, the python image library and opencv libraries have used for the calibration module. Only a prototype of the SfM procedures are implemented using Matlab(some of the functions are from Peter Kovesi[13]) and

[4]The scene points should be in front of the camera

python and the 3D point clouds are visualized using Meshlab sofware. The problems regarding to the implementation of this work was mainly related to SfM part because of the necessity of the usage of different libraries. The problems are raised when we wanted to embed the prototype of SfM module, as we did for calibration module, inside choregraphe because of some incomparability issues of array manipulation libraries such as numpy[27]. However, the SfM module can still be used with Matlab using the Matlab SDK of Aldebaran robotics.

One advantage of having a procedure as a module in choregraphe is that, it is an encapsulated module that provides abstraction for any user. Therefore, by using the choregraphe boxes any user can obtain results even without knowledge of methods.

Camera calibration test results are verified using widely used and recognized camera calibration toolbox[2] by the computer vision society, written by Jean-Yves Bouguet. First we have tested the camera calibration module fixing NAO and moving pattern in different rotation and translations to be sure that; camera calibration module is working and the programmed movements are not a limitation for the camera calibration procedure(the movement of Nao might be limited compared to the former rotations and translations). Therefore, we have to apply two different tests. The first one verifies that the implementation of the method has no errors. The second one verifies the calibration can also be done moving Nao. The images are shown in figure 6 are for the first test.

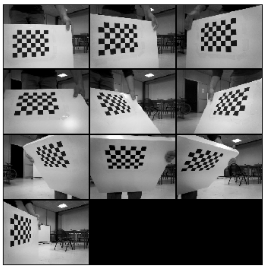

Fig. 6. The images of the chessboard pattern in different rotations and translations

The results for intrinsic parameters of the camera by calibration toolbox is

$$K = \begin{bmatrix} 556.686 & 0.0 & 328.683 \\ & 557.590 & 231.765 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

The results for intrinsic parameters of the camera by our method is

$$K = \begin{bmatrix} 557.473 & 0.0 & 329.959 \\ 0.0 & 558.239 & 232.031 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

The results show the intrinsic parameters of the camera with both method is the same. Since the test condition to be verified for the first test is to obtain the same results with the calibration toolbox, we can verify that the method that we have implemented works.

The second test set is the images captured by fixing the calibration pattern and moving NAO. The image set is shown and the movements are illustrated in figure 7. The results of our method for intrinsic parameters is

$$K = \begin{bmatrix} 574.608 & 0.0 & 320.613 \\ 0.0 & 575.934 & 230.169 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

As the results show that the calibration method works and the recovered intrinsic parameters are correct with a very slight change which can be ignored. Note that the more image of patterns we get the more accurate results we will obtain.

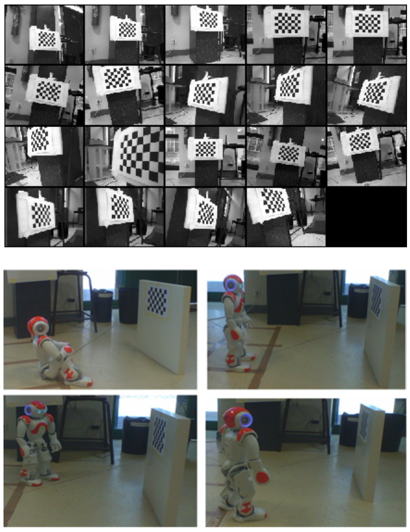

Fig. 7. The images acquired for calibration and some poses of Nao.

Structure from Motion module is tested for 3 different set of scene structures. In order to obtain a good 3D reconstruction we needed as much correspondence points between two sequential images as we can extract and match, thus the number of feature points detection and matching was crucial. The first test images are acquired from outdoor scenes(8).

Fig. 8. Two views of an outdoor scene. The green dots are representing the matched feature points.

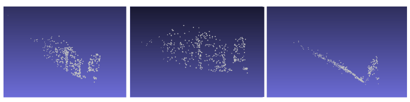

Fig. 9. The 3D point cloud of the library(8) from three different point of view.

The second one is a pre-prepared environment with a lot of feature points and specific and solid structure(10).

Fig. 10. An pre-prepared indoor environment with a lot of feature points.

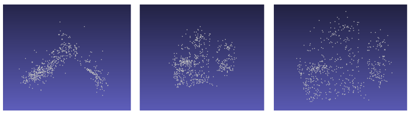

Fig. 11. The 3D reconstruction of the scene shown in Figure 10.

The third test set is the random sequence of images that have no pre-prepared structure.(12).

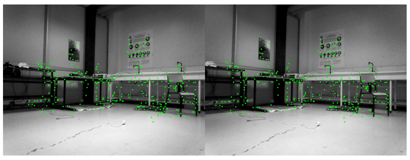

Fig. 12. A random scene.

Fig. 13. The 3D reconstruction of the scene in Figure 12.

The results of 3D reconstructions are shown above. Generally we had true reconstructions except when NAO was moving in an environment where there was less number of future points. Scenes like those are not easy to reconstruct and the method is not well responded to such environments. Since, there is no ground-truth of such applications we can visually inspect the results from the created point clouds as shown in figures. Moreover The results of SfM also verifies that the calibration module results are correct. Obviously, with wrong camera parameters the results of 3D reconstructions would not be true.

V. Conclusions and Future Work

The goal of this paper was to explain the developed methods that would allow humanoid robot Nao to calibrate it's camera and obtain a 3D reconstruction of the scene from two views and recover the pose of the camera. Also in terms of reusability we aimed to design the system for both calibration and Sfm in a way that it provides high level of abstraction.

In terms of reusability and modularity of the applications, for both considerations of future developments and ease of use, we have developed the calibration module as a stand alone package that any user, even without the knowledge of the details of the method, can employ this module and obtain the calibration results for all the works that require camera characteristics. The SfM module is developed as an external script instead of as a module of Nao.

In this project all the goals that are defined in the problem statement are achieved. The contribution of this work is the application of challenging tasks; camera calibration and structure from motion using a mobile robot(Nao).

There are variety of computer vision algorithms that can be applied on Nao using our work as basis. Some of those may be listed as future works;

- The first future work of this thesis would be importing the SfM method we have already prototyped and tested as external script.
- A dense 3D reconstruction application may be developed over this work using more than two images as input for the scene and applying different correspondence algorithms.
- An Incremental SfM methods might be employed using our method as a basis in a way that each and every image points are registered to the initial point cloud. Moreover a real-time SfM might be an application to extend this work.
- A stand alone Robot localization and navigation application can be developed using the motion and structure information.

References

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008.

[2] Jean-Yves Bouguet. Camera calibration toolbox. 2000.

[3] Stphane Christy and Radu Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:1098–1104, 1996.

[4] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.

[5] Olivier D. Faugeras, Quang-Tuan Luong, and Stephen J. Maybank. Camera self-calibration: Theory and experiments. In *Proceedings of the Second European Conference on Computer Vision*, ECCV '92, pages 321–334, London, UK, UK, 1992. Springer-Verlag.

[6] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[7] Chris Harris and Mike Stephens. volume 15, pages 147–151. Manchester, UK, 1988.

[8] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[9] Richard I. Hartley. Self-calibration from multiple views with a rotating camera. pages 471–478. Springer-Verlag, 1994.

[10] Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, June 1997.

[11] David W. Jacobs. Linear fitting with missing data for structure-from-motion. *Computer Vision and Image Understanding*, 82:2001, 1997.

[12] Ismael Kassem. Structure from motion and camera self-calibration. In *Master Thesis, Master of Science in Vision and Robotics*, University of Burgundy.

[13] Peter Kovesi. Matlab and octave functions. 2012.

[14] H. C. Longuet-Higgins. Readings in computer vision: issues, problems, principles, and paradigms. chapter A computer algorithm for reconstructing a scene from two projections, pages 61–62. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[15] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[16] Shyjan Mahamud, Martial Hebert, Yasuhiro Omori, and Jean Ponce. Provably-convergent iterative methods for projective structure from motion, 2001.

[17] Gerard Medioni and Sing Bing Kang. *Emerging Topics in Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.

[18] J More. *The Levenberg-Marquardt algorithm: implementation and theory*, volume 630, pages 105–116. Springer, 1978.

[19] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion using local bundle adjustment. *Image Vision Comput.*, 27(8):1178–1193, July 2009.

[20] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, June 2004.

[21] John Oliensis, Senior Member, Richard Hartley, and Senior Member. Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. In *In ECCV (4*, pages 214–227, 2006.

[22] Srikumar Ramalingam, Suresh K. Lodha, and Peter Sturm. A generic structure-from-motion framework. *Comput. Vis. Image Underst.*, 103(3):218–228, September 2006.

[23] Fabio Remondino and Clive Fraser. Digital camera calibration methods: considerations and comparisons. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 36(5):266–272, 2006.

[24] Peter Sturm and Bill Triggs. A factorization based algorithm for multi-image projective structure and motion. pages 709–720. Springer-Verlag, 1996.

[25] Carlo Tomasi. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9:137–154, 1992.

[26] R Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

[27] www.python.org. Python documantation. 2012.

[28] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, November 2000.