Robot-assisted Distal Locking of Long Bone Intramedullary Nails: Localization, Registration, and In-Vitro Experiments

Ziv Yaniv and Leo Joskowicz

School of Engineering and Computer Science
The Hebrew University of Jerusalem, Jerusalem 91904, Israel.
Email: {zivy,josko}@cs.huji.ac.il

March 31, 2004

Abstract

We have developed an image-guided robot-based system to assist orthopedic surgeons in performing distal locking of long bone intramedullary nails. The system consists of a bone-mounted miniature robot fitted with a drill guide that provides rigid mechanical guidance for hand-held drilling of the distal screws' pilot holes. The robot is automatically positioned so that the drill guide and nail distal locking axes coincide using a single fronto-parallel fluoroscopic X-ray. This paper describes new methods for accurate and robust drill guide and nail hole localization and registration and reports the results of in-vitro system accuracy experiments.

1 Introduction

The growing demand for precise, minimally invasive surgical interventions is driving the search for ways to use computers in conjunction with advanced assistance devices to improve surgical planning and execution. Over the past decade, a variety of Computer Integrated Surgery (CIS) systems have been designed, mostly for neurosurgery, laparoscopy, maxillofacial surgery, and orthopedics [20]. Recent studies are starting to show the clinical benefits of these systems.

CIS systems can potentially benefit many orthopedic surgical procedures, including total hip and total knee replacement, pedicle screw insertion, fracture reduction, and ACL ligament reconstruction. These procedures are ubiquitous and high volume in operating rooms worldwide. They involve rigid bone structures that image well,

require preoperative planning, and employ instruments and tools, such as implants, screws, drills, and saws that require precise positioning. Indeed, a few dozen CIS systems for these procedures are currently deployed [25].

Our work focuses on a technique for fracture reduction called closed intramedullary nailing. Closed intramedullary nailing is currently the routine procedure of choice for reducing fractures of the femur and the tibia [4]. It restores the integrity of the fractured bone by means of a nail inserted in the medullary canal. The nail is inserted without surgically exposing the fracture through an opening, usually in the proximal part of the bone. The surgeon reduces the fracture by manipulating the proximal and distal bone fragments through the leg until they are aligned. The surgeon then inserts a guide wire, reams the canal if necessary, and drives the nail in. In most cases, the surgeon also inserts lateral proximal and distal interlocking screws to prevent fragment rotation and bone shortening. The procedure is performed under X-ray fluoroscopy, which is used to view the position of bone fragments, surgical tools, and implants. Numerous X-ray fluoroscopic images are required, especially during distal locking [28].

Distal locking – the insertion of lateral screws to prevent nail rotation – has long been recognized as one of the most challenging steps in this procedure. Since the nail deforms by several millimeters to conform to the bone canal shape, the exact position of the distal locking nail holes' axes cannot be determined in advance. By repeatedly alternating between anterior-posterior and lateral X-ray fluoroscopic views, the surgeon adjusts the entry point and orientation of the drill so that its axis coincides with the corresponding nail hole axis. Drilling proceeds incrementally, with each advance verified with a new pair of X-ray fluoroscopic images (Figure 1). Once the pilot hole passing through the distal locking nail's hole has been drilled, the locking screw is fastened. Complications include inadequate fixation, malrotation, bone cracking, cortical wall penetration, and bone weakening due to multiple or enlarged pilot holes. The literature reports that the surgeon's direct exposure to radiation per procedure without the use of CIS systems is 3-30 minutes, of which 31-51% is spent on distal locking depending on the patient anatomy and the surgeon's skill [28].

2 Previous work

Many non-CIS devices have been developed for distal locking [22]. Examples include proximally mounted targeting devices, stereo fluoroscopy, mechanical guides, and optical and electro-magnetic navigation systems that help locate the center of the distal locking nail holes. However, all of these devices and techniques have deficiencies: they are only selectively applicable, are cumbersome and difficult to use, or are not sufficiently accurate, and thus fail to significantly reduce the likelihood of patient complications.

Fluoroscopy-based CIS navigation systems [14, 15, 18] take the guesswork out of targeting. The systems enhance, reduce, or altogether eliminate X-ray fluoroscopic

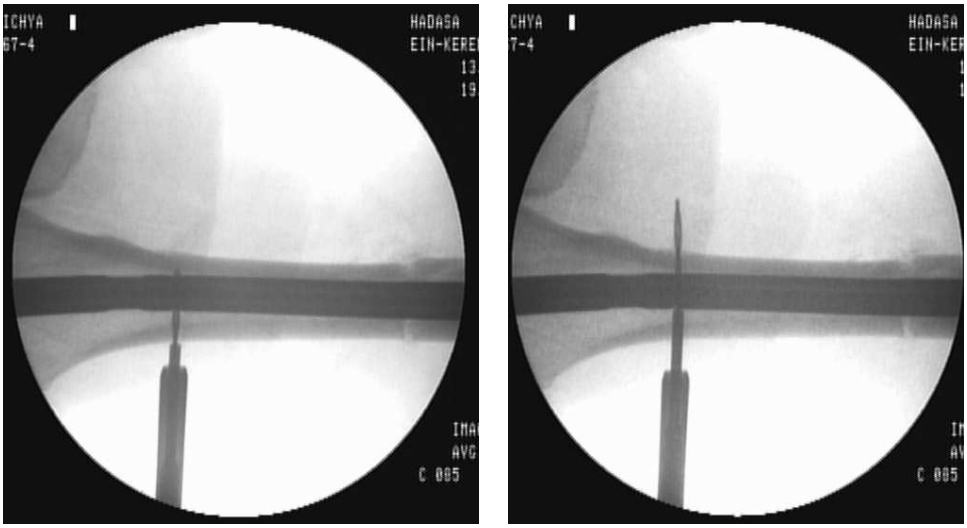
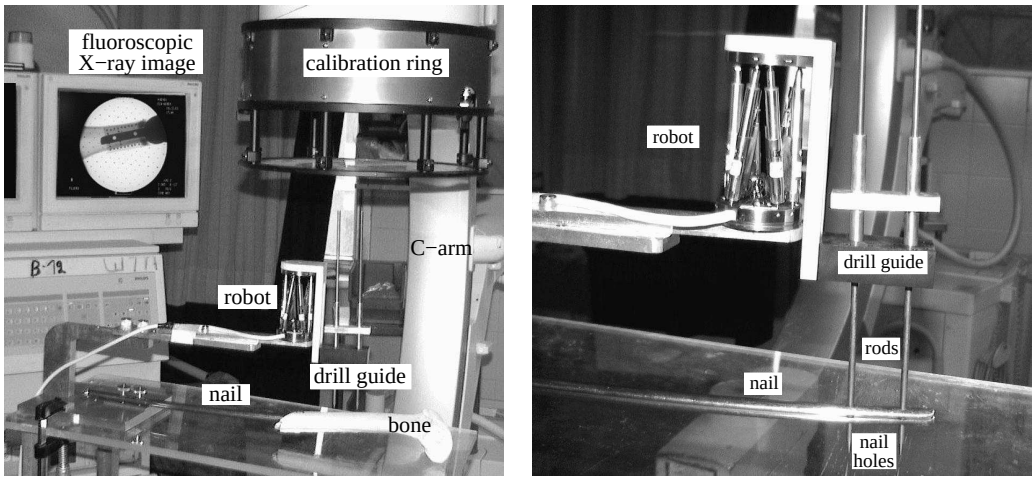


Figure 1: X-ray fluoroscopic images acquired during pilot hole drilling.

images by replacing them with a virtual reality view in which the bone and instruments' positions are continuously updated and viewed on-screen as they move. They help the surgeon align the drill axis with the distal locking nail hole axis to an accuracy of about 1mm and 1^o. However, they do not provide a mechanical guide for the hand-held drill, which can slip or deviate from its planned trajectory as the drilling proceeds. Thus, the surgical outcomes are still largely dependent on the surgeon's skill.

Robot-based CIS systems are designed to assist the surgeon in implementing the preoperative plan by mechanically positioning and sometimes executing the surgical action itself [5, 29]. The robots are either adapted floor-standing industrial robots, or table-mounted custom-designed serial robots. They are usually voluminous and heavy despite the fact that they have relatively small workloads and workvolumes. In these systems, bone immobilization or real-time dynamic tracking is required, since the relative configuration of the bone with respect to the robot must be known precisely at all times. This complicates the registration procedure and adversely affects the overall system accuracy. A very recent novel development is the miniature parallel robot MARS [27]. The miniature robot is directly mounted onto the patient, forming a single rigid body with the anatomy. This removes the need for anatomy immobilization or real time tracking during surgery. We propose to use this robot to automatically align a drill guide with the distal nail hole axes providing accurate mechanical guidance for manual drilling [19].



(a) (b)

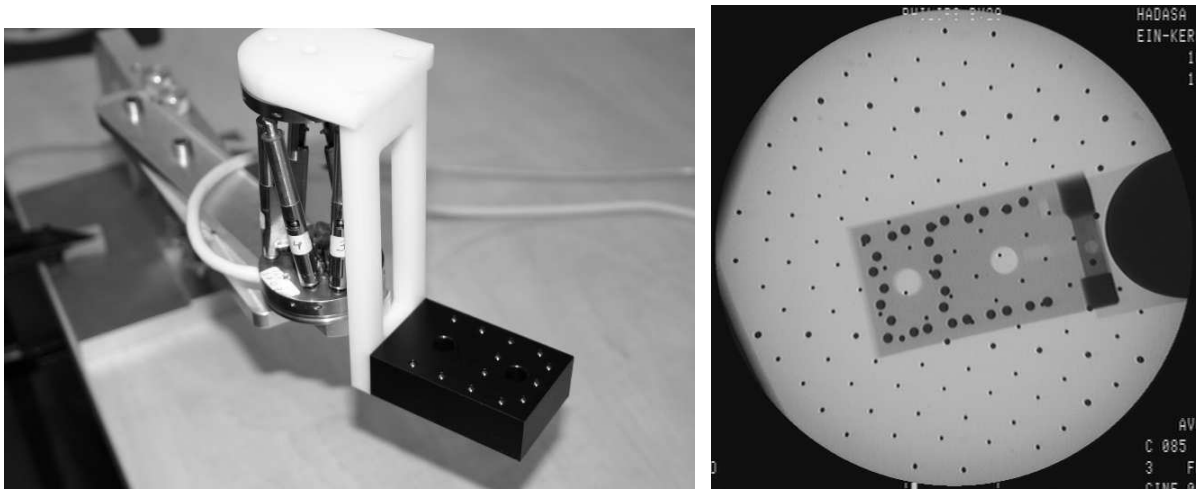
Figure 2: Photographs of (a) the in-vitro setup and (b) the MARS robot in its desired configuration, where the drill guide and nail holes axes are aligned, as shown by the two rods passing through them.

3 System concept

The proposed system consists of a miniature patient-mounted robot, a drill guide, an image calibration ring for the fluoroscopic X-ray unit, a PC computer with a video frame grabber, and image based guidance software.

MARS is a 5x5x7cm^3, 150–gram six-degree-of-freedom parallel manipulator whose work volume is about 10cm^3 and whose accuracy is better than 0.1mm. When locked, it is rigid and can withstand forces of a few kilograms. It is mounted either directly on the bone or on the nail head (Figure 2). Both options are minimally invasive and eliminate the need for leg immobilization or real-time tracking during surgery. The drill guide is a Delrin block with two guiding holes 30mm apart (the spacing between the nail holes). It has a pattern of 28 3mm stainless steel fiducial spheres asymmetrically distributed on two planes 20mm apart that are used for its spatial localization (Figure 3).

The surgical protocol is as follows. Once the fracture has been reduced and the nail has been inserted the image calibration ring is mounted on the C-arm image intensifier. The surgeon then mounts the robot onto the patient, either directly onto the bone, or onto the nail head. Our software then estimates the required robot pose so that the drill guide axes and nail axes coincide. This estimation can be done in one of two modes, interactive or fully automatic. In the interactive mode the X-ray technician orients the C-arm, guided by our software, so that it is in a fronto-parallel setting with the nail holes. This mode requires the acquisition of a fluoroscopic image for every prospective orientation. In the fully automatic mode a single fluoroscopic image is acquired and the required robot pose is estimated. Once



(a) (b)

Figure 3: (a) drill guide target mounted on robot and (b) its fluoroscopic X-ray image.

the robot is positioned the surgeon manually drills the pilot holes through the robot mounted drill guide, removes the robot, and completes the surgery according to the standard protocol.

Accurate and robust computation of the transformation that aligns the drill guide and the nail hole axes is a challenging image processing and pose estimation task. Localization of the nail holes and the drill guide is difficult because partial occlusions are inherent to the setup (the robot is mounted close to the nail holes and the image includes the nail, bone, and soft tissue). The nail holes are small ($5mm$ diameter, about 20 pixels), nearby ($30mm$), and appear as ellipses in the images, so the accuracy with which their axes can be determined is limited. Furthermore, only one fluoroscopic X-ray image can be used, since there is no tracking of the C-arm pose. Finally, the C-arm imaging system exhibits orientation-dependent distortions and internal imaging parameters variations. To cope with these challenges, we have developed a novel model-based approach for robust and accurate localization of the drill guide target and nail holes and for the registration of their axes.

We model the fluoroscopic camera as a pin-hole camera with distortion, as this has been shown to be an appropriate approximation of the X-ray imaging process[3, 15, 23, 32]. In previous work, we describe a robust automatic C-arm calibration algorithm that includes fiducial localization, distortion correction and camera calibration [23]. The algorithm computes the distortion correction and camera calibration parameters from an X-ray fluoroscopic image in three steps. First, it locates the fiducials' projections and pairs them with their spatial location. Next, it computes the distortion correction parameters, and then computes the calibration parameters. Accurate and robust localization of the fiducials and their pattern is the most important step, since all parameters critically depend on it. Our experiments show that submillimetric accuracy for the combined dewarping and camera calibration is achievable even when

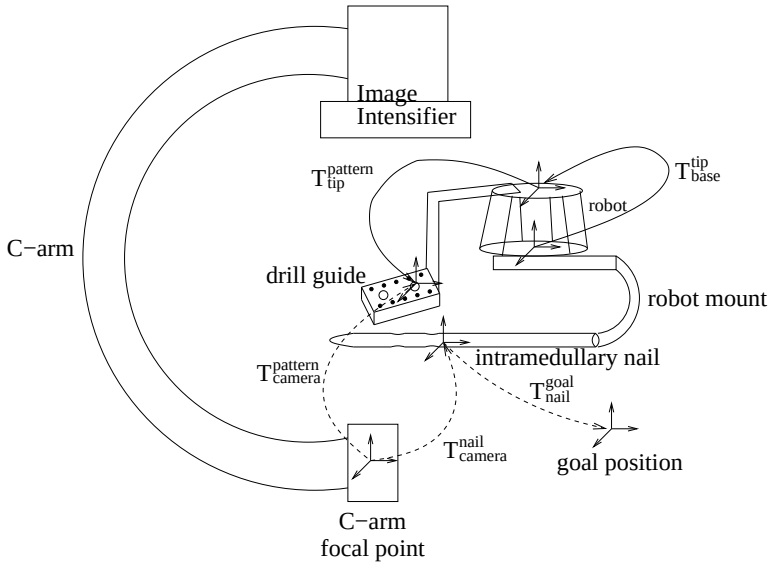


Figure 4: Robot alignment registration chain. Solid lines mark known transformations, dashed lines mark computed transformations.

only 60% of the fiducials are detected.

The nail is modeled as a planar object with two circular holes. This is a good approximation as the nail's distance from the camera focal point is between 450-700mm and the nail diameter is 10-15mm (45 to 70 times smaller than the distance).

To align the robot so that the drill guide axes coincide with the nail hole axes we need to close the transformation chain depicted in Figure 4:

$$T_{base}^{goal} = T_{base}^{tip} T_{tip}^{pattern} (T_{camera}^{pattern})^{-1} T_{camera}^{nail} T_{nail}^{goal}$$

where T_{base}^{tip} and $T_{tip}^{pattern}$ are known from design and $T_{camera}^{pattern}$ and T_{camera}^{nail} are computed from the C-arm internal camera parameters and the fluoroscopic image, and T_{nail}^{goal} is computed from the previous two transformations.

Our method consists of three steps: 1) C-arm distortion correction and calibration; 2) drill guide and nail hole identification; 3) drill guide and nail pose estimation. We describe the later two next.

4 Drill guide and nail hole identification

4.1 Drill guide identification

Drill guide identification is performed by first detecting the circular fiducials in the image and then finding the correct correspondence with their 3D model. The key issues are handling partial occlusions and missing fiducials, and minimizing positional inaccuracies.

The target fiducials are detected in two steps: 1) localization and 2) circle fitting. Localization is performed using a modified circle Hough transform [17] and a model-based analysis of the transform accumulator. Since the circular fiducials are darker than the image background, the Hough transform voting scheme is constrained so that edge pixels will only cast their vote if the vector connecting them to the hypothesized circle center is in the opposite direction of the gradient at the edge location. The contents of the transform accumulator are examined to identify the $k > 28$ circles which received the most votes. Considering a few more candidate circles is necessary since the accumulator may contain multiple peaks for the same fiducial (these are higher than the peaks for other fiducials). The algorithm then computes the average radius and number of votes of the five circles with the most votes and selects all circles whose radius is within ± 2 pixels of the average radius and which have received more than half of the average votes. These circles, with very high probability, belong to the target.

The selected set of circles may contain overlapping circles, which are due either to multiple responses for the same fiducial or to fiducial overlap. In our imaging setup, there can be at most two overlapping fiducials. Thus, for each set of overlapping circles, we only retain the two circles with the highest number of votes. These pairs of overlapping circles correspond either to two overlapping fiducials or to a single fiducial. When the overlap area between the pair of circles is larger than 60% of the circle's area, it is a single circle, otherwise it is two. Circle fitting is performed using the Random Sample Consensus (RANSAC) paradigm [7]. For each circle, the algorithm collects all edge elements contained in its circumscribing square and fits a circle to them. This removes the dependency on the Hough transform discretization accuracy.

The correspondence between the detected fiducials and their 3D model is computed using homographies. Correctly pairing the detected fiducials and the 3D model fiducials is difficult, as there are always missing fiducials which are occluded by the nail. To overcome these problems, we use the geometry of the drill guide target: we use lines instead of points, since lines are less sensitive to partial occlusions.

Since the fiducials are distributed on two planes, the goal is to find the pair of homographies which minimizes the distance between the detected circles and the result of applying the homographies to the target fiducials. The pair of homographies is computed in three steps:

1. Find line with maximal support (either long or short target axis), the other axis is nearly perpendicular to the one found.

2. Find all lines which contain two spheres and are nearly perpendicular (parallel) to the long axis. Sort them according to their mean projection onto the long (short) axis. The next step requires that at least two lines per target plane were identified. This means that according to the target design for the long axis there are $4 - 7$ lines and for the short axis $4 - 5$ lines. If more than the maximal number of lines are detected then the identification step has failed.

3. Go over all $\binom{7}{k}, \binom{5}{j}, 4 \leq k \leq 7, 4 \leq j \leq 5$ line pairing options. For each pairing compute the two homographies corresponding to the target planes using the direct linear transformation method [13]. Each pairing is ranked according to the number of target-image point matches and the sum of distances. We transform all target points and pair them with the closest image point which is at most $2r$ pixels away (where r is the average radius of the detected circles). The best pairing is the one which maximizes the number of matches, and if there is more than one such pairing we take the one which minimizes the total distance between transformed target points and their matched image points.

We now have the target-image correspondences. These may contain pairings between several target points and a single image point. This happens when the target spheres overlap in the image and only one is detected. All such multiple pairings are discarded. Finally a visual validation is done, overlaying the image with the detected circles and the paired target spheres. Figure 5 shows the stages of drill guide target detection and identification, with Figure 5(f) used for visual validation.

4.2 Nail hole identification

The location of the distal locking nail holes in the X-ray fluoroscopic image is determined by first locating the nail's longitudinal contour and then locating its holes from their expected position with respect to the contour (Figure 6).

To locate the nail longitudinal contours, we use a 3D Hough transform in which the nail is modeled as a band consisting of two parallel lines with a known distance between them. The Hough transform voting scheme is constrained so that pixels which are on parallel lines will only cast their vote if the gray level values between them are lower than the gray level values outside the band.

The search for the nail holes is then performed on the pixels contained between the nail's contours. The algorithm sweeps a parallelepiped window whose sizes are equal to the nail width along the nail's medial axis. The two locations containing the maximal number of edge elements correspond to the locations of the distal locking nail holes. The algorithm then fits an ellipse to the edge data contained inside the parallelepipeds. The edge elements originate from the nail holes, the drill guide target and the C-arm calibration target (Figure 6(b)). The ellipse parameter estimation must take this into account.

We considered two approaches to cope with the outlying data: 1) RANSAC, and 2) a model-based approach. The RANSAC approach is a general randomized framework for dealing with outlying data. In our context subsets of five pixels are randomly chosen and the ellipse parameters are estimated using a generic conic fitting solution [33]. For each such estimate the consensus set is found and the largest set is used as input for a least squares estimation procedure. Edge elements are incorporated into the consensus set only if their geometric distance from the ellipse is below a predefined threshold. The model-based approach uses only edge elements which belong

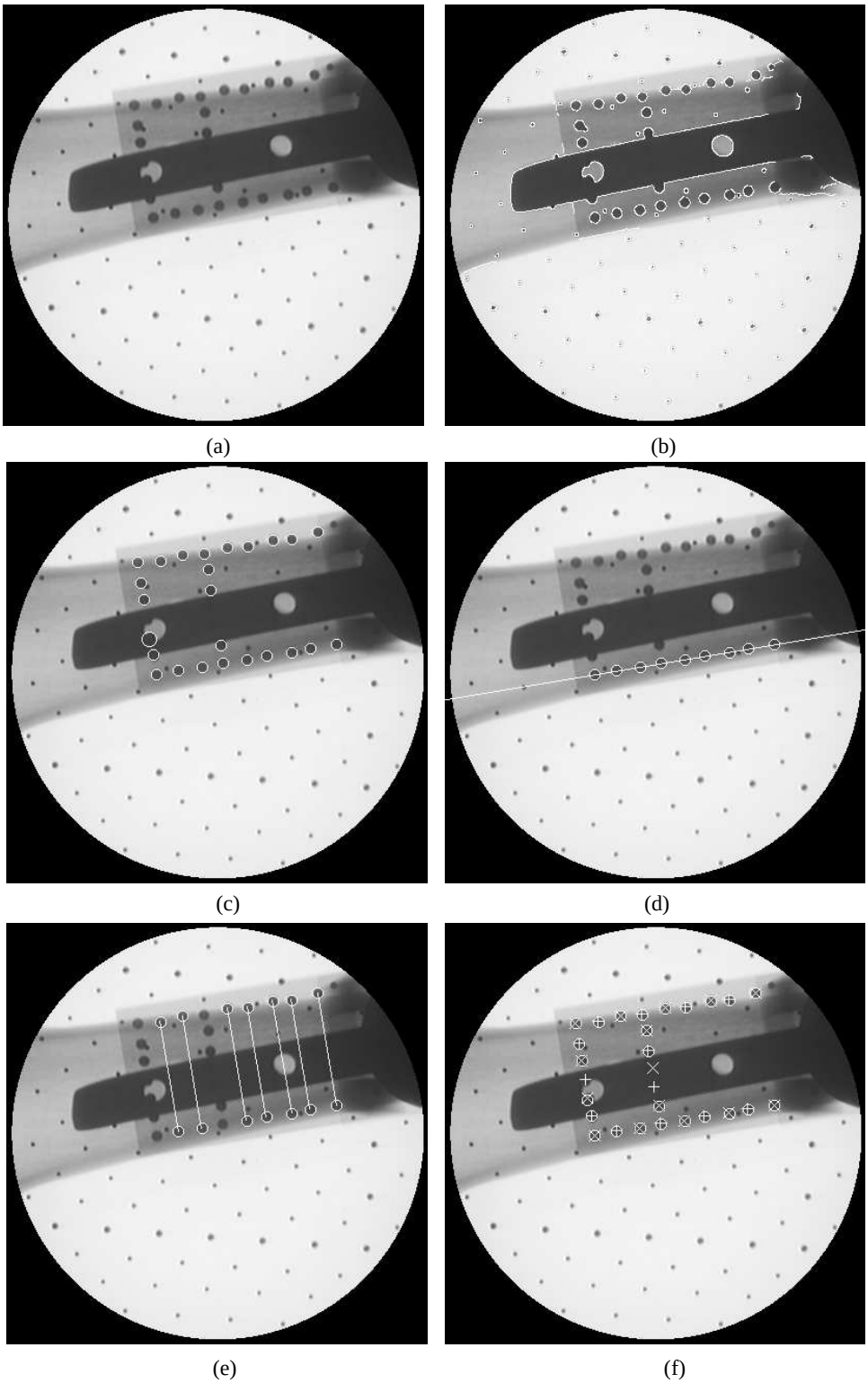


Figure 5: Drill guide target identification: (a) distortion corrected image (b) result of edge detection overlayed on previous image (c) detected fiducials (d) line with maximal support (e) lines which have a support of two fiducials (f) white circles mark detected fiducials and + and × denote associated model planes.

to the convex hull of the set of elements. This is because ellipses are convex shapes and because the nail is opaque, so outlying edges can only be present in the interior of the ellipse. In both approaches we estimated the final ellipse parameters using a non-linear geometric least squares optimization (Downhill-Simplex), initialized with the result of an algebraic least squares estimate [8, 12].

After experimenting with both approaches we opted for the model-based one. The RANSAC approach failed occasionally due to the threshold value for incorporation into the consensus set. Figure 6 shows an example of the nail hole identification using the model-based approach. Final evaluation of the ellipse fitting is done by visual inspection.

5 Drill guide and nail pose estimation

The desired robot alignment is:

$$\widehat{T}_{base}^{tip} = T_{base}^{goal}(T_{tip}^{pattern})^{-1}$$

where

$$T_{base}^{goal} = T_{base}^{tip} T_{tip}^{pattern} (T_{camera}^{pattern})^{-1} T_{camera}^{nail} T_{nail}^{goal}$$

and we have three unknown transformations $T_{camera}^{pattern}$, T_{camera}^{nail} and T_{nail}^{goal} whose computation is described in the following subsections.

In our context, we perform point based pose estimation, a problem which has been thoroughly studied in computer vision. We explored several solutions to this problem with the following factors in mind, ordered according to their importance: 1) achieve the highest accuracy possible; 2) deal with noisy input data without any outliers; 3) compute solution within several seconds.

We evaluated four algorithms, Direct Linear Transform (DLT) [6], Depth Based (DB) [1], Genetic Algorithm (GA) and, Non-Linear optimization (NL). Empirically we found that the classic photogrammetric approach of Non-Linear optimization initialized with the results of the DLT gave the best results both in terms of time and accuracy. The algorithms are described in Appendix A. A detailed comparison is provided in Section 6.

5.1 Drill guide pose estimation

The drill guide pose is computed by non-linear minimization of the projection distances between the known fiducial projection coordinates (x_i, y_i) and the expected ones $(\widehat{x}_i, \widehat{y}_i)$:

$$\mathbf{v}^* = \arg\min_{\mathbf{v}} 0.5 \left(\sum_{i=1}^{n} (x_i - \widehat{x}_i(\mathbf{v}))^2 + (y_i - \widehat{y}_i(\mathbf{v}))^2 \right)$$

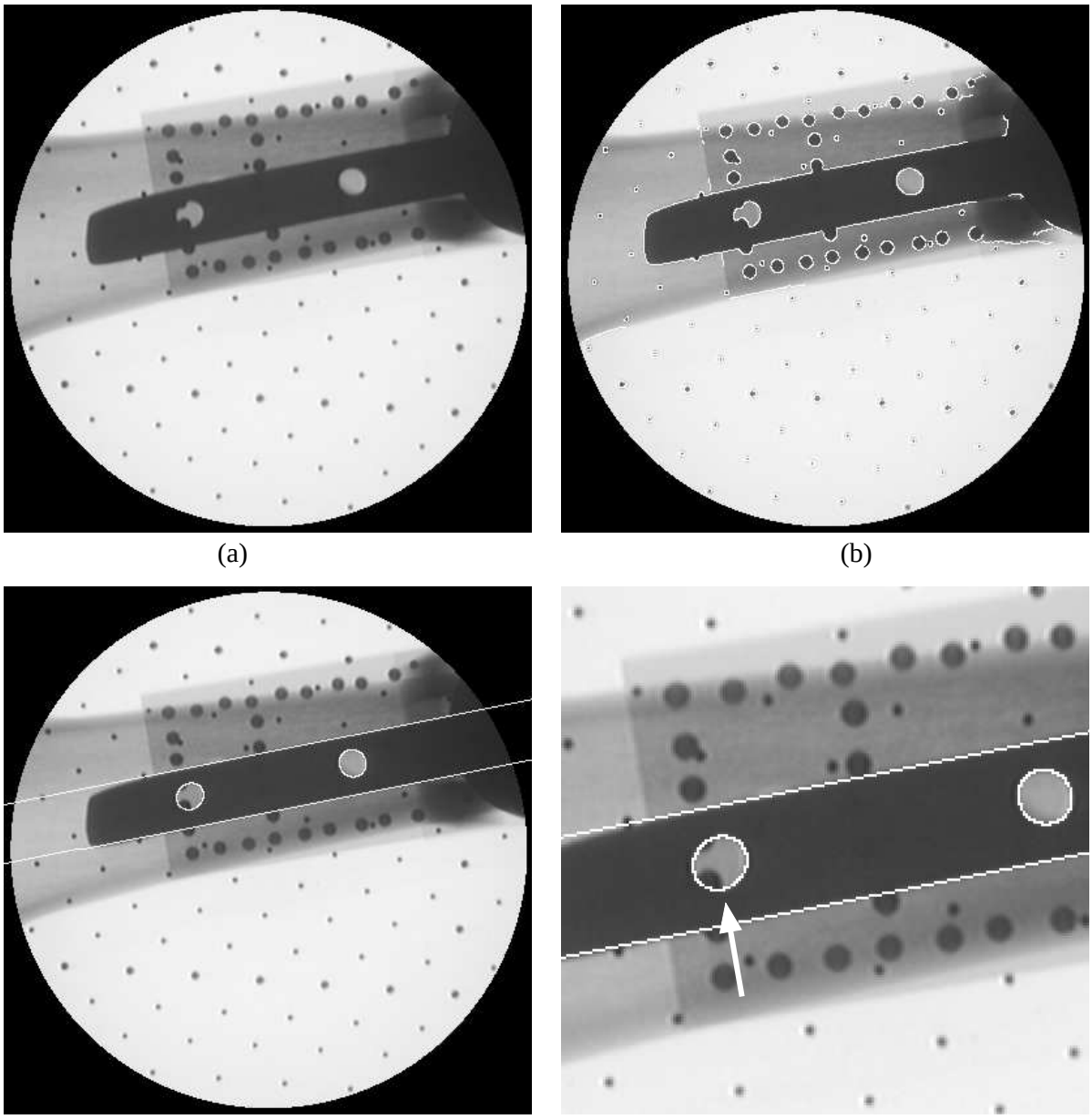


Figure 6: Nail hole detection:(a) distortion corrected image (b) result of edge detection overlayed on previous image (c) nail's longitudinal contour and detected ellipses (d) zoom in on detected nail holes an arrow indicates occlusions due to drill guide spheres.

where $\mathbf{v}$ is the rigid transformation parameterization. The non-linear minimization is performed with the Levenberg-Marquardt method as described in [24].

5.2 Nail pose estimation

We experimented with two approaches which determine the nail pose. The first approach is an interactive one, requiring the X-ray technician to acquire a few fluoroscopic images until the orientation of the C-arm forms a fronto-parallel setup with the nail holes. The second approach is more generic and requires the acquisition of a single fluoroscopic image. Although the second approach is computationally more attractive we have empirically found the first approach to be more accurate.

5.2.1 Interactive nail pose determination

In this approach we interactively guide the user to a known pose, a fronto-parallel setup between the C-arm and nail holes. To arrive at this setup, the X-ray technician images the nail in several orientations which are scored on two criteria, hole circularity and deviation angle between hole supporting plane and camera viewing direction. When the setup is a fronto parallel setup the nail holes should appear in the image as circles and the angular deviation between the nail hole's supporting plane normal and the camera viewing direction is zero.

We compute the measure of hole circularity as the aspect ratio of the ellipse which is fit to the data points, and the angular deviation between the computed supporting plane normal [10, 21] and the camera's viewing direction[1]. This serves to guide the X-ray technician to the desired setup. We have empirically found this process to require up to six images.

Once the fronto-parallel setup is achieved, the nail distance z from the camera focal point is estimated. Using the average diameter of the two nail holes in the image d_i and their known diameter d_w we have $z = \frac{f d_w}{d_i}$ and the nail hole location is computed accordingly.

$$\mathbf{t} = \begin{bmatrix} \frac{z}{f}\mathbf{p}_x \\ \frac{z}{f}\mathbf{p}_y \\ z \end{bmatrix}$$

where $\mathbf{p}$ and $\mathbf{q}$ are the centers of the two circles. The nail rotation relative to the camera is:

$$R = \begin{bmatrix} \mathbf{p}_x - \mathbf{q}_x & \mathbf{p}_y - \mathbf{q}_y & 0 \\ \mathbf{p}_y - \mathbf{q}_y & \mathbf{q}_x - \mathbf{p}_x & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

[1] The computation of the supporting plane normal has two solutions yielding two angles. We display the smaller angle as we assume that the images are acquired near the fronto-parallel setup.

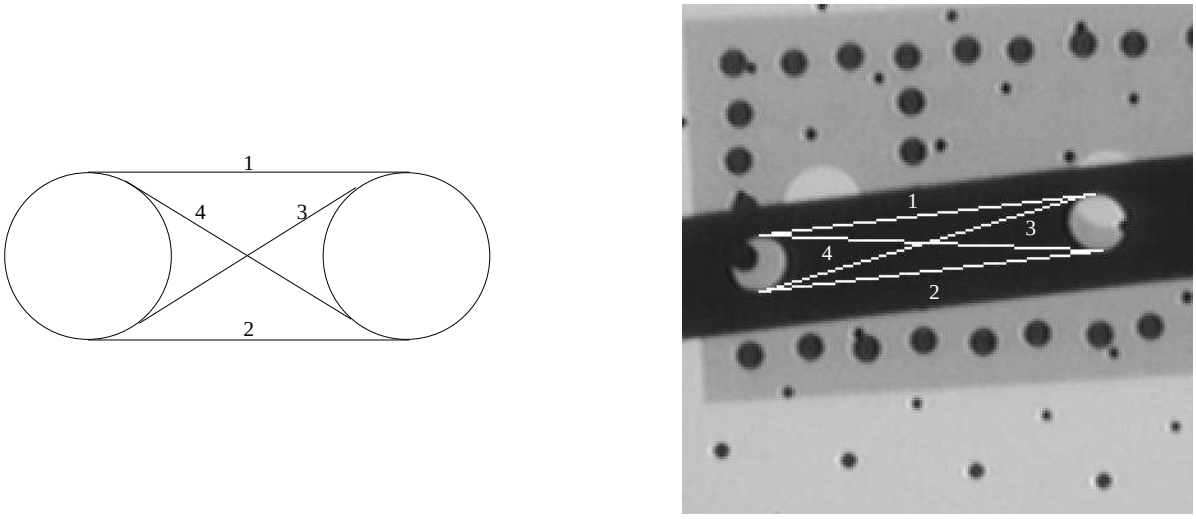


Figure 7: Bitangents in model and image. Intersection point of bitangents three and four is used in the evaluation of the bitangent identification step.

The nail's X axis direction (choice of $\mathbf{p}, \mathbf{q}$) is set so that its angular deviation from the direction of the drill guide's X axis is minimal.

5.2.2 Generic nail pose determination

To determine the nail pose from a single image acquired from an unknown pose we find points on the two ellipses which we are able to identify with their three dimensional model. Given four or more such points we solve the point-based pose estimation problem as described in Appendix A.

Given a pair of circles which have undergone a projective transfromation we would like to find a set of points which are invariant to this transformation. In general the intersection and bitangent points of the two circles are invariant under a projective transformation [10, 26]. In our specific case we know that the two ellipses have four bitangent lines and no intersection points. The bitangent lines are computed as described in [26].

In homogeneous coordinates the ellipses are:

$$\mathbf{x}^T Q_1 \mathbf{x} = 0 \qquad \text{and} \qquad \mathbf{x}^T Q_2 \mathbf{x} = 0$$

The normal to Q_1 is $\mathbf{n_1} = 2Q_1\mathbf{x}$ and the tangent line at $\mathbf{x_1}$ is the set of points r which are orthogonal to $\mathbf{n_1}$:

$$\mathbf{n_1}^T \mathbf{r} = 0$$
$$2\mathbf{x_1}^T Q_1 \mathbf{r} = 0$$

Let $\mathbf{l_1}^T = \mathbf{x_1}^T Q_1$ then $\mathbf{l_1}^T \mathbf{r} = 0$ is a line. Using the duality of points and lines we can view the line $\mathbf{l_1}^T$ as a point on the ellipse Q_1^{-1}:

$$\mathbf{l^1}^T Q_1^{-1} \mathbf{l^1} = \mathbf{x_1}^T Q_1 Q_1^{-1} Q_1 \mathbf{x_1} = 0 \quad (1)$$

Likewise for the second ellipse we have:

$$\mathbf{l_2}^T Q_2^{-1} \mathbf{l_2} = 0 \quad (2)$$

The bitangent $\mathbf{l}$ of Q_1, Q_2 satisfies both Equations 1 and 2. Hence $\mathbf{l}$ is given by the intersections of the ellipses Q_1^{-1} and Q_2^{-1}, the four solutions to the two polynomial equations defined by the ellipses.

The order of the tangency points around the ellipse does not change as we are dealing with a perspective transformation. This means there are four possible matches due to each of the bitangent lines. The matches are then ranked using homographies. The bitangents are enumerated in counter clockwise order according to the bitangent points on the first circle (Figure 7). As the nail is modeled as a planar object with two circular holes, every prospective match induces a homography. The homography is computed using the four bitangent points associated with the first circle, and it is ranked according to the distance between the intersection point of bitangents three and four in the image and the mapping of their intersection point from the world plane to the image plane. The match which has the minimal distance is the correct one.

Once the correct match is obtained, there are thirteen pairs of points on the nail hole plane and the image plane. These points are the eight bitangent points and the five intersection points of the bitangents. Finally, we also compute the image coordinates of the nail hole centers as described in [21]. This gives us a total of fifteen point matches for use in the pose estimation.

5.3 Goal pose estimation

We define the goal pose, the final pose of the drill guide target, using the drill guide target pose, $T_{camera}^{pattern}$, and the nail pose, T_{camera}^{nail}. The orientation of the goal pose is the same as that of the nail. The location of the goal pose is defined as the intersection point of the nail pose Z axis with a plane which is parallel to the camera xy plane and at the same distance as the drill guide target from the camera origin (Figure 8).

6 Experimental results

We conducted four sets of experiments to quantify the accuracy and robustness of the proposed system.

1. Evaluation of the image processing algorithms.
2. Evaluation of the point-based pose estimation algorithms.

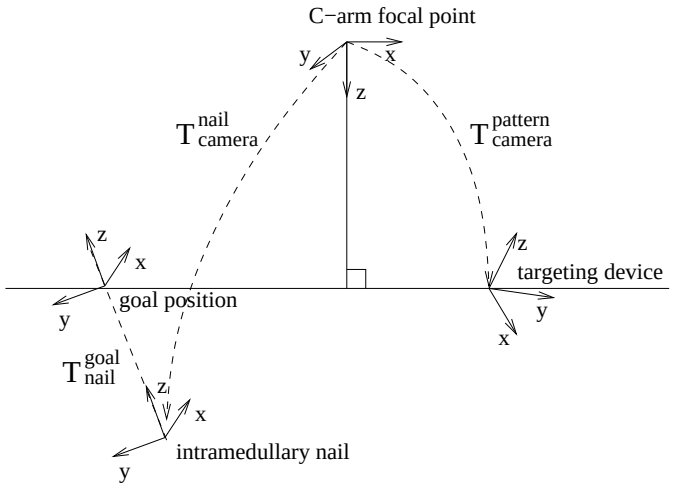


Figure 8: The goal pose is defined by the drill guide target and nail poses in the camera coordinate system.

3. Assessment of variability of the robot reference pose.

4. Whole-system accuracy assessment.

All images were acquired with a 9" BV29 C-arm (Phillips, The Netherlands). They were 768x576 pixels, 8-bit gray-scale. All algorithms were implemented in C++ and ran on a 2.4 GHz, 1GB RAM PC running Windows XP. Processing times were 5–10 seconds per image.

1. Image processing evaluation

To test the robustness of the drill guide and nail hole identification we manually placed the robot and C-arm in random positions. We then acquired 78 fluoroscopic X-ray images. The drill guide and nail holes were correctly identified in 61 images. There were six failures due to problems with the drill guide target detection, because not enough target fiducials were found. In 11 images the procedure failed due to severe nail occlusion of the target. These failures can be disregarded as the operator can manually orient the robot so that more of the fiducials are visible. There were no false positives, the detection was correct in all images for which our software stated that the drill guide and nail holes were detected. All failures were also detected automatically by the software.

2. Point-based pose estimation evaluation

The following point-based pose estimation algorithms were evaluated:

- DLT - Direct Linear Transform [6].
- DB - Depth Based estimation [1].
- GA - Genetic Algorithm.

- NL - Non-Linear optimization (Levenberg-Marquardt).

The first two algorithms, DLT and DB, do not require initialization. The GA approach requires coarse parameter bounds, which were computed using the camera internal parameters and the knowledge that the imaged object is between the focal point and the image plane. The NL method requires initialization which is obtained from the application of the three other methods, NL-DLT, NL-DB, and NL-GA. Different rotation parametrizations (unit quaternion and Euler angles) did not affect the results of the NL method.

Table 1 summarizes the results of pose estimation on 61 input images. For each image automatic drill guide detection was performed and its output was used as input for pose estimation. The number of input points varied between 20 to 28. To compare the algorithms we choose the NL method as a reference, as it gave the smallest 2D and 3D errors in all experiments.

The NL method gave the most accurate results and converged to the same minimum no matter which initialization method was used. It is followed by DB, GA and finally DLT. The DLT method gave the worst results as expected, since it ignores the constraints on the rotation matrix in addition to the noisy input. The GA method did worse than the DB method most likely because of early termination with sub-optimal results. Both the GA and DB methods are not accurate enough for our purposes. They are not sensitive enough to changes in the distance along the viewing direction. This is most evident in the DB method where the 2D and 3D errors are below $0.5mm$, yet the difference along the viewing direction from the reference pose may be up to $10mm$.

The NL-DLT method and DLT method tie for the best running time. The GA method's running times are consistent and are similar for all inputs. The DB method gave the worst running times. This is because one of the steps involved in the computation is a Singular Value Decomposition of a matrix whose number of rows is a cubic function of the number of input points. This leads to longer computation times and a large standard deviation, which reflects the size of the input ranging from 20 to 28 points.

In conclusion, the NL-DLT gives the best results both in accuracy and time.

3. Robot reference pose evaluation

We define the reference robot pose as a pose in which two rods ($4mm$ diameter) pass through the drill guide and nail holes ($5mm$ diameter), Figure 2(b). This guarantees successful locking. As this definition admits many poses we also need comparison measures for the different poses. Comparing robot poses using the difference in Euler angles and translations does not reflect the relevant errors for the task at hand. Instead, we look at the angular deviation of the drill guide axes and the distance between axes entry and exit points. This distance is defined as the in-plane distance between the intersection points of the axes and two planes located at 100mm and 120mm from the robot base (the nail is located between these planes). Figure 9

algorithm	time (sec) mean std	error (mm) mean std		deviation from NL-GA (mm,degrees) mean std					
		2D	3D	x	y	z	θ_x	θ_y	θ_z
DLT	0.00	409.35	229.06	180.68	140.04	48.16	1.62	0.97	0.07
	0.00	184.86	95.86	78.32	61.39	49.74	1.64	0.89	0.14
DB	7.49	0.26	0.16	0.16	0.11	4.76	0.16	0.26	0.01
	4.44	0.34	0.22	0.31	0.13	6.27	0.23	0.48	0.01
GA	4.24	1.34	0.87	0.37	1.07	4.66	1.02	0.88	2.35
	0.16	2.79	1.85	0.70	2.43	10.79	1.90	1.82	5.34
NL-DLT	0.00	0.09	0.06	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.10	0.06	0.00	0.00	0.00	0.00	0.00	0.00
NL-DB	7.54	0.09	0.06	0.00	0.00	0.00	0.00	0.00	0.00
	4.47	0.10	0.06	0.00	0.00	0.00	0.00	0.00	0.00
NL-GA	4.32	0.09	0.06	0.00	0.00	0.00	0.00	0.00	0.00
	0.22	0.10	0.06	0.00	0.00	0.00	0.00	0.00	0.00

Table 1: Comparison of pose estimation algorithms (mean errors and standard deviations computed for 61 data sets).

describes this setup.

To assess the possible variability of the reference pose we manually placed the robot in 18 different poses in which the two rods passed through the drill guide and nail holes. We then compared all the transformations to each other and found an angular variability in the range of $0^o - 0.9^o$ and a translational variability in the entry and exit points in the range of $0 - 3.9mm$.

4. Whole system evaluation

To quantify the accuracy of the whole system we conducted the following experiment, Figure 2(a). The robot was first placed in a reference pose. Next, under the guidance of our software, we oriented the C-arm so that it formed a fronto-parallel setup with the nail holes. We then placed the robot in 17 random poses, acquired fluoroscopic X-ray images and computed the robot alignment for each pose. Drill guide target and nail holes were correctly detected in all images. We now have two options for computing the nail pose, assume a fronto-parallel setup (section 5.2.1) or without any assumptions (section 5.2.2). We then computed the robot pose and compared it to the reference pose.

When assuming a fronto-parallel setup our results showed a mean angular error of 1.3^o (std $= 0.4^o$) between drill guide axes, and a mean $3.0mm$ error (std $= 1.1mm$) in the entry and exit points. When no assumptions were made our results showed a mean angular error of 2.7^o (std $= 2.1^o$) between drill guide axes, and a mean $3.4mm$ error (std $= 1.2mm$) in the entry and exit points. Figure 10 shows a projection of the drill

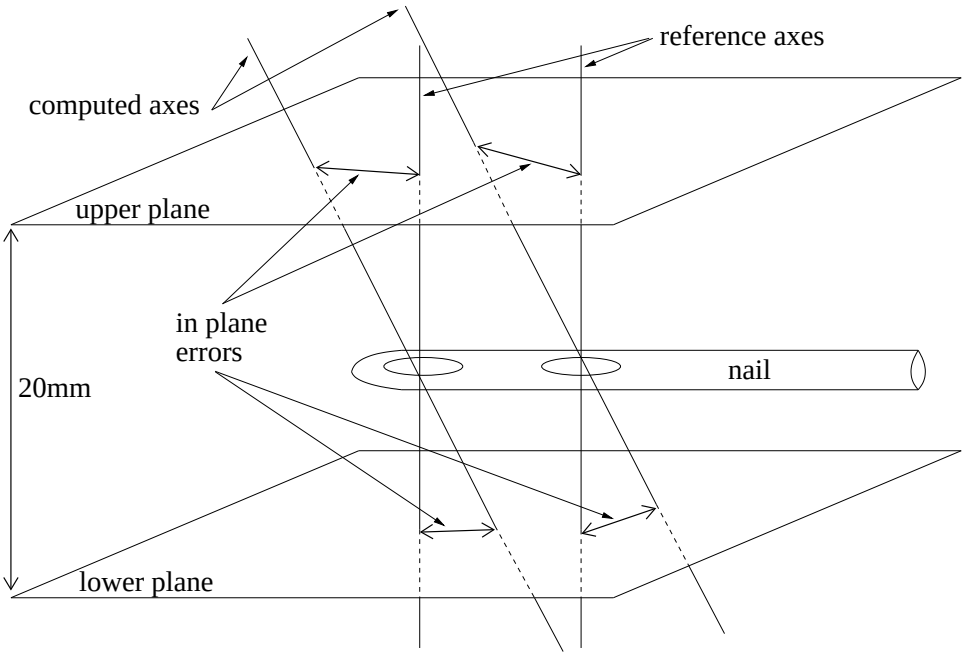


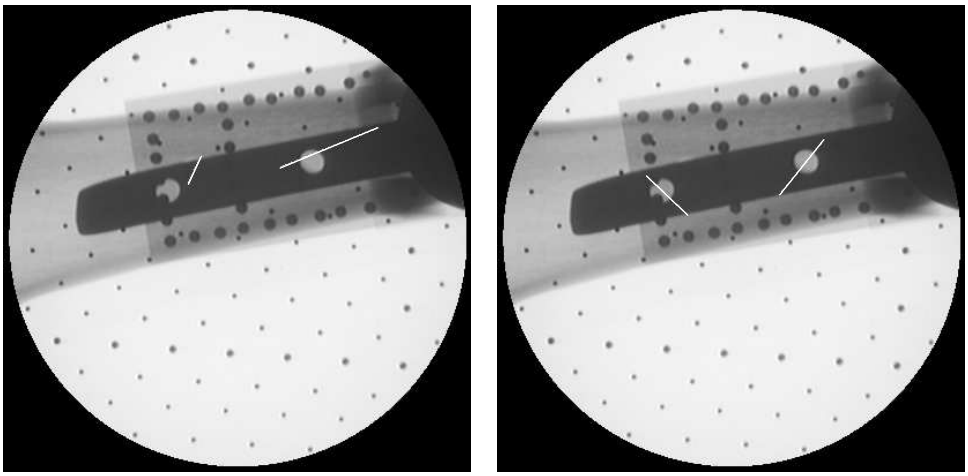
Figure 9: The in-plane distance between computed axes intersection points and reference axes intersection points.

guide axes before and after robot alignment. Figure 11 summarizes the results for all 17 experiments. The translational error is either comparable between approaches or smaller when using the fronto-parallel assumption. The rotational error is smaller when using the fronto-parallel assumption and has a smaller variability.

We conclude that the interactive alignment of the C-arm is more robust than pose estimation using a single image. In both approaches the nail holes are detected and ellipses fit to them. When using the interactive approach we indicate the deviation from the fronto-parallel setup by displaying the ellipse aspect ratio and the angular deviation between the circle supporting plane normal and the camera's viewing direction. This computation is done separately for each of the ellipses and in the case of the aspect ratio, only two of the ellipse parameters are used in the computation. In the automatic pose estimation all ellipse parameters from both ellipses are used in a single computation, making the computation more sensitive to parameter errors from both ellipses.

7 Conclusions

We have presented a robust, automatic method for aligning the drill guide and the nail holes axes based on a few fluoroscopic X-ray images. The method is part of a new image-guided miniature robot-based system to assist orthopedic surgeons in performing distal locking of long bone intramedullary nails.



(a) (b)

Figure 10: Projection of drill guide axes onto image (a) in original robot pose, angular difference from reference pose is 0.0^o, translational difference for entrance and exit points is $8.5mm$ (b) in computed robot pose, angular difference from reference pose is 1.7^o, translational difference for entrance and exit points is $1.5mm$

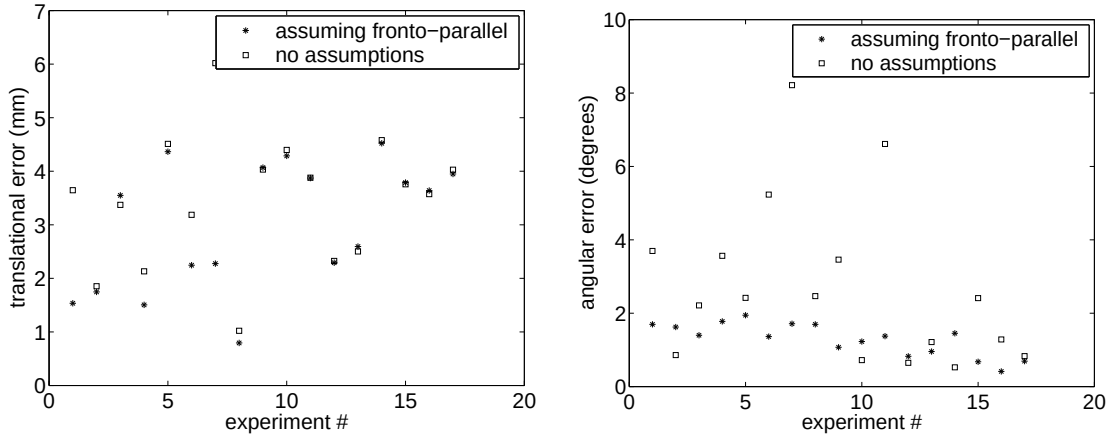


Figure 11: Angular and translational errors relative to reference pose.

As part of the robot alignment two approaches to nail pose determination were developed. The first approach is an interactive one in which the X-ray technician aligns the C-arm and nail in a fronto-parallel setup using image based guidance. The second approach estimates the nail pose from a single image of the nail holes. Our experimental results show that the interactive approach is more accurate then the automatic one.

Using the interactive approach we achieve a mean angular error of 1.3^o (std $= 0.4^o$) between the computed drill guide axes and the actual locking holes axes, and a mean $3.0mm$ error (std $= 1.1mm$) in the entry and exit drill point, which is adequate for successfully locking the nail.

We are currently working on mechanical and algorithmic improvements to increase the accuracy of the system, and will then proceed with a cadaver and in-vivo evaluation.

A Point based pose estimation

Given a camera whose internal parameters are known, a set of points in an object coordinate system and a corresponding set of image points, compute the camera pose (rotation and translation).

We use the pin-hole camera model described in [6], with the following intrinsic parameters:

f – camera focal length.

u_0, v_0 – image origin coordinates at the intersection of the optical axis and the image plane.

k_u, k_v – horizontal and vertical pixel scale factors in the image plane.

and projection matrix:

$$K = \begin{bmatrix} \alpha_v & 0 & u_0 \\ 0 & \alpha_u & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where $\alpha_u \equiv -fk_u$ and $\alpha_v \equiv -fk_v$.

A.1 Direct linear transform (DLT)

For every point pair $\mathbf{p}, \widehat{\mathbf{p}}$ we have $\widehat{\mathbf{p}} = M\mathbf{p}$, where $M = KT$ and K is a projection matrix (Equation 3) and $T = [R; t]$ is a 4×4 rigid transformation matrix. In homogenous coordinates $\widehat{\mathbf{p}} = [\widehat{p}_x, \widehat{p}_y, \widehat{p}_w]^T$ and $M\mathbf{p}$ are not necessarily equal but they have the same direction, that is:

$$\widehat{\mathbf{p}} \times M\mathbf{p} = 0$$

and in explicit form:

$$\begin{bmatrix} \widehat{p}_y \mathbf{m_3 p} - \widehat{p}_w \mathbf{m_2 p} \\ \widehat{p}_w \mathbf{m_1 p} - \widehat{p}_x \mathbf{m_3 p} \\ \widehat{p}_x \mathbf{m_2 p} - \widehat{p}_y \mathbf{m_1 p} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

where $\mathbf{m_{1\ldots3}}$ are the rows of M.

This gives us three equations from each point pair:

$$\begin{bmatrix} 0^T & -\widehat{p}_w \mathbf{p}^T & \widehat{p}_y \mathbf{p}^T \\ \widehat{p}_w \mathbf{p}^T & 0^T & -\widehat{p}_x \mathbf{p}^T \\ -\widehat{p}_y \mathbf{p}^T & \widehat{p}_x \mathbf{p}^T & 0^T \end{bmatrix} \begin{bmatrix} \mathbf{m_1} \\ \mathbf{m_2} \\ \mathbf{m_3} \end{bmatrix} = 0$$

As these equations are linearly dependent we can take the first two equations from each point pair. Stacking these equations we get a least squares formulation $min\|A\mathbf{m}\|$

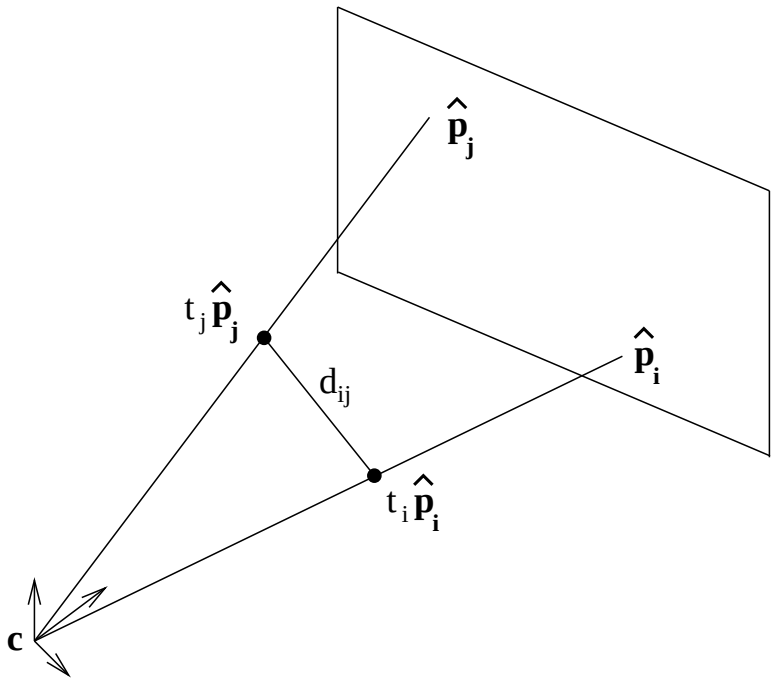


Figure 12: Geometric constraint relating projected points $\widehat{\mathbf{p}}_\mathbf{i}$, $\widehat{\mathbf{p}}_\mathbf{j}$, the distance between the original points in the world d_{ij} and the associated scale factors t_i, t_j along the projection rays.

and for a meaningful solution we need to add a constraint. Two possible constraints are $\|\mathbf{m}\| = 1$ and $\|m_{31}, m_{32}, m_{33}\| = 1$ (last row of rotation matrix is orthonormal). Finally, the rigid transformation parameters are extracted from the projection matrix [6].

This formulation ignores the constraints on the rotation matrix (i.e. that $R^T R = I$ and $det(R) = 1$). The resulting transformation will usually be a projective transformation, but not a perspective transfromation. Where we expect a rigid transformation we will have an affine transformation. To overcome this problem we take the rotation matrix as the "closest" rotation to the matrix we have (via SVD).

A.2 Linear, depth based estimation

This section describes the linear point based method of Ansar and Daniilidis [1]. This is a closed form solution which utilizes all n available points.

The depth of every point in space along the ray emanating from the camera focal point and going through its corresponding $2D$ projection is computed. Once the depths are available the point coordinates in the camera coordinate system are readily computed and we can solve the exterior orientation problem using either quaternions [16] or rotation matrices [2, 30].

Let $\widehat{\mathbf{p_i}} = [x_i, y_i, f]$ and $\widehat{\mathbf{p_j}} = [x_j, y_j, f]$ be two points on the image plane, where f is the camera focal length. These points correspond to two points which are on rays emenating from the camera focal point. Let t_i and t_j be two positive scale factor such that $t_i\widehat{\mathbf{p_i}}$ and $t_j\widehat{\mathbf{p_j}}$ are these $3D$ points. The distance d_{ij} between these points is known and yields the following constraint (Figure 12):

$$\begin{aligned} d_{ij}^2 &= \|t_i\widehat{\mathbf{p_i}} - t_j\widehat{\mathbf{p_j}}\|^2 \\ &= (t_i\widehat{\mathbf{p_i}} - t_j\widehat{\mathbf{p_j}})^T(t_i\widehat{\mathbf{p_i}} - t_j\widehat{\mathbf{p_j}}) \\ &= t_i^2\widehat{\mathbf{p_i}}^T\widehat{\mathbf{p_i}} + t_j^2\widehat{\mathbf{p_j}}^T\widehat{\mathbf{p_j}} - 2t_it_j\widehat{\mathbf{p_i}}^T\widehat{\mathbf{p_j}} \end{aligned} \quad (4)$$

Every pair of points contributes an equation of the form of Equation 4. We now linearize the above equation by defining $t_it_j \equiv t_{ij}$ such that each point contributes an equation of the form:

$$t_{ii}\widehat{p_{ii}} + t_{jj}\widehat{p_{jj}} - 2t_{ij}\widehat{p_{ij}} - d_{ij}^2\rho = 0$$

where

$$\begin{aligned} \widehat{p_{ii}} &\equiv \widehat{\mathbf{p_i}}^T\widehat{\mathbf{p_i}} \\ \widehat{p_{jj}} &\equiv \widehat{\mathbf{p_j}}^T\widehat{\mathbf{p_j}} \\ \widehat{p_{ij}} &\equiv \widehat{\mathbf{p_i}}^T\widehat{\mathbf{p_j}} \end{aligned}$$

and ρ is an additional variable which allows us to represent the equations as a homogeneous equation system (subject to the constraint $\rho = 1$):

$$M\mathbf{t} = 0 \quad (5)$$

where[2]

$$M = \left[\begin{array}{cccc|cccccc} -2\widehat{p_{12}} & 0 & \cdots & 0 & \widehat{p_{11}} & \widehat{p_{22}} & 0 & \cdots & \cdots & -d_{12}^2 \\ 0 & -2\widehat{p_{13}} & \cdots & 0 & \widehat{p_{11}} & 0 & \widehat{p_{33}} & 0 & \cdots & -d_{13}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & -2\widehat{p_{n-2,n-1}} & 0 & 0 & \cdots & \widehat{p_{n-2,n-2}} & \widehat{p_{n-1,n-1}} & 0 & -d_{n-2,n-1}^2 \\ 0 & \cdots & 0 & -2\widehat{p_{n-1,n}} & 0 & \cdots & 0 & \widehat{p_{n-1,n-1}} & \widehat{p_{nn}} & -d_{n-1,n}^2 \end{array}\right]$$

[2]Given n points the dimensions of the matrix M are $[\frac{n(n-1)}{2}, \frac{n(n+1)}{2} + 1]$ and according to its explicit form it has a null space of dimension $\frac{n(n+1)}{2} + 1 - \frac{n(n-1)}{2} = n + 1$.

and

$$\mathbf{t} = \begin{bmatrix} t_{12} \\ \vdots \\ t_{1n} \\ t_{23} \\ \vdots \\ t_{2n} \\ \vdots \\ t_{n-1,n} \\ t_{11} \\ \vdots \\ t_{nn} \\ \rho \end{bmatrix}$$

If the null space of M is one dimensional then the solution of Equation 5 is computed using the Singular Value Decomposition of M. Given $M = U\Sigma V^T$ take the column of V corresponding to the smallest singular value and normalize this vector by its last entry, imposing the constraint $\rho = 1$. We now have the solution to Equation 5 up to a uniform sign error, which is resolved by the constraint that the scale factors be positive.

As the dimension of M's null space is $n+1$, the solution is obtained by re-imposing the quadratic nature of the original problem. Since the solution belongs to the null space of M there exist real numbers λ_i such that

$$\mathbf{t} = \sum_{i=1}^{n+1} \lambda_i \mathbf{v_i} \tag{6}$$

For any integers $\{i, j, k, l\}$ and any permutation $\{i', j', k', l'\}$ we have $t_{ij}t_{kl} = t_{i'j'}t_{k'l'}$. We impose the quadratic constraints of the form $t_{ii}t_{jk} = t_{ij}t_{ik}$ [3] on the right hand side of Equation 6 yielding constraints on the coefficients λ_i of the form:

$$\begin{aligned} t_{ii}t_{jk} - t_{ij}t_{ik} &= \sum_{a=1}^{n+1} \lambda_{aa}(v_a^{ii}v_a^{jk} - v_a^{ij}v_a^{ik}) + \\ &\quad \sum_{a=1}^{n+1}\sum_{b=a+1}^{n+1} \lambda_{ab}(v_a^{ii}v_b^{jk} + v_b^{ii}v_a^{jk} - v_a^{ij}v_b^{ik} - v_b^{ij}v_a^{ik}) \\ &= 0 \end{aligned}$$

where $\lambda_{ab} \equiv \lambda_a\lambda_b$.

[3] There are $n^2(n-1)/2$ such constraints. All constraints of the form $t_{ii}t_{jk} = t_{ij}t_{ik}$ where $i = j$ or $i = k$ are not used as they yield rows of zeros.

We now have the following homogeneous system of equations in the $(n+1)(n+2)/2$ variables λ_{ab}:

$$K\lambda = 0$$

where each constraint contributes a row in K of the form:

$$\begin{bmatrix} v_1^{ii}v_2^{jk} + v_2^{ii}v_1^{jk} - v_1^{ij}v_2^{ik} - v_2^{ij}v_1^{ik} \\ \vdots \\ v_1^{ii}v_{n+1}^{jk} + v_{n+1}^{ii}v_1^{jk} - v_1^{ij}v_{n+1}^{ik} - v_{n+1}^{ij}v_1^{ik} \\ \vdots \\ v_n^{ii}v_{n+1}^{jk} + v_{n+1}^{ii}v_n^{jk} - v_n^{ij}v_{n+1}^{ik} - v_{n+1}^{ij}v_n^{ik} \\ v_1^{ii}v_1^{jk} - v_1^{ij}v_1^{ik} \\ \vdots \\ v_{n+1}^{ii}v_{n+1}^{jk} - v_{n+1}^{ij}v_{n+1}^{ik} \end{bmatrix}^T$$

and

$$\lambda = \begin{bmatrix} \lambda_{12} \\ \vdots \\ \lambda_{1,n+1} \\ \lambda_{23} \\ \vdots \\ \lambda_{2,n+1} \\ \vdots \\ \lambda_{n,n+1} \\ \lambda_{11} \\ \vdots \\ \lambda_{n+1,n+1} \end{bmatrix}$$

The null space of K is one dimensional (see [1] for details) and is computed using the SVD of K. Now that we have the solution λ up to a scale factor we compute the solution to Equation 5 as:

$$\mathbf{t} = \sum_{i=1}^{n+1} \lambda_{1i}\mathbf{v_i}$$

and then normalize $\mathbf{t}$ by its last entry to enforce the constraint $\rho = 1$. Using the scale factors t_i we compute the point coordinates in the camera coordinate system and solve the pose estimation problem using Horn's quaternion based solution.

A.3 Non-linear method

We consider two representations for the rotational part of the rigid transformation a rotation matrix given in Euler angles and a unit quaternion.

The exterior orientation is formulated as the following nonlinear least squares optimization problem:

$$\mathbf{v}^* = \arg\min_{\mathbf{v}} 0.5 \left(\sum_{i=1}^{n} e_{xi}^2 + \sum_{i=1}^{n} e_{yi}^2 \right) \tag{7}$$

where:

- $\mathbf{v}$ - rigid registration parameterization:
  - $\mathbf{v} = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]$ - Translations and Euler angles.
  - $\mathbf{v} = [t_x, t_y, t_z, q_x, q_y, q_z]$ - Translations and vector part of unit quaternion.
- Errors in x and y coordinates:

$$\begin{aligned} e_x(\mathbf{v})_i &= x_i - \widehat{p_{xi}} \\ e_y(\mathbf{v})_i &= y_i - \widehat{p_{y_i}} \end{aligned} \tag{8}$$

Optimization is performed with the Levenberg-Marquardt method, as described in [24]. This requires the computation of the gradient and approximated Hessian which are given by [9]:

$$\begin{aligned} \mathbf{g} &= J\mathbf{f} \\ H &= JJ^T \end{aligned}$$

where:

$$\mathbf{f}_{2n\times 1} = \begin{bmatrix} e_{x1} \\ e_{y2} \\ \vdots \\ e_{xn} \\ e_{yn} \end{bmatrix} \quad J_{6\times 2n} = \begin{bmatrix} \frac{\partial \mathbf{f}_1}{\partial t_x} & \cdots & \frac{\partial \mathbf{f}_{2n}}{\partial t_x} \\ \frac{\partial \mathbf{f}_1}{\partial t_y} & & \frac{\partial \mathbf{f}_{2n}}{\partial t_y} \\ \frac{\partial \mathbf{f}_1}{\partial t_z} & & \frac{\partial \mathbf{f}_{2n}}{\partial t_z} \\ \frac{\partial \mathbf{f}_1}{\partial \theta_x} & & \frac{\partial \mathbf{f}_{2n}}{\partial \theta_x} \\ \frac{\partial \mathbf{f}_1}{\partial \theta_y} & & \frac{\partial \mathbf{f}_{2n}}{\partial \theta_y} \\ \frac{\partial \mathbf{f}_1}{\partial \theta_z} & \cdots & \frac{\partial \mathbf{f}_{2n}}{\partial t_x} \end{bmatrix}$$

A.3.1 Rotation as an Euler angles matrix

Given a rigid transformation represented by a translation vector $\mathbf{t}$ and an Euler angle rotation matrix R ($c \equiv \cos$ and $s \equiv \sin$) we have:

$$
\begin{aligned}
R(\theta_x, \theta_y, \theta_z) &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \\
&= \begin{bmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix} \\
&= \begin{bmatrix} c_z c_y & c_z s_y s_x - s_z c_x & c_z s_y c_x + s_z s_x \\ s_z c_y & s_z s_y s_x + c_z c_x & s_z s_y c_x - c_z s_x \\ -s_y & c_y s_x & c_y c_x \end{bmatrix}
\end{aligned}
$$

After transforming the point to the camera coordinate system it is projected onto the image plane by the projection matrix K (Equation 3) yielding the image point $\widehat{\mathbf{p}}$:

$$
\begin{aligned}
\widehat{p_x} &= \frac{(\alpha_u \mathbf{r_1} + u_0 \mathbf{r_3})\mathbf{p} + \alpha_u t_x + u_0 t_z}{\mathbf{r_3}\mathbf{p} + t_z} \\
\widehat{p_y} &= \frac{(\alpha_v \mathbf{r_2} + v_0 \mathbf{r_3})\mathbf{p} + \alpha_v t_y + v_0 t_z}{\mathbf{r_3}\mathbf{p} + t_z}
\end{aligned}
\tag{9}
$$

To apply the Levenberg-Marquardt optimization we differentiate the error terms $e_x(\mathbf{v})$ and $e_y(\mathbf{v})$ with respect to $\mathbf{v} = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]$.

Defining:

$$
\begin{aligned}
d &\equiv \mathbf{r_3} p + t_z \\
n_x &\equiv (\alpha_u \mathbf{r_1} + u_0 \mathbf{r_3})\mathbf{p} + \alpha_u t_x + u_0 t_z \\
n_y &\equiv (\alpha_v \mathbf{r_2} + v_0 \mathbf{r_3})\mathbf{p} + \alpha_v t_y + v_0 t_z
\end{aligned}
$$

We differentiate the error terms e_x, e_y given in Equation 8 with respect to $\mathbf{v} = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]$:

$$
\begin{aligned}
\frac{\partial e_x}{\partial t_x} &= \frac{\alpha_u}{d} \\
\frac{\partial e_x}{\partial t_y} &= 0 \\
\frac{\partial e_x}{\partial t_z} &= \frac{u_0}{d} - \frac{n_x}{d^2} \\
\frac{\partial e_x}{\partial \theta_x} &= \frac{p_y(\alpha_u \frac{\partial r_{12}}{\partial \theta_x} + u_0 \frac{\partial r_{32}}{\partial \theta_x}) + p_z(\alpha_u \frac{\partial r_{13}}{\partial \theta_x} + u_0 \frac{\partial r_{33}}{\partial \theta_x})}{d} - \frac{n_x}{d^2}\left(p_y \frac{\partial r_{32}}{\theta_x} + p_z \frac{\partial r_{33}}{\theta_x}\right) \\
\frac{\partial e_x}{\partial \theta_y} &= \frac{p_x(\alpha_u \frac{\partial r_{11}}{\partial \theta_y} + u_0 \frac{\partial r_{31}}{\partial \theta_y}) + p_y(\alpha_u \frac{\partial r_{12}}{\partial \theta_y} + u_0 \frac{\partial r_{32}}{\partial \theta_y}) + p_z(\alpha_u \frac{\partial r_{13}}{\partial \theta_y} + u_0 \frac{\partial r_{33}}{\partial \theta_y})}{d} - \\
&\quad \frac{n_x}{d^2}\left(p_x \frac{\partial r_{31}}{\theta_y} + p_y \frac{\partial r_{32}}{\theta_y} + p_z \frac{\partial r_{33}}{\theta_y}\right) \\
\frac{\partial e_x}{\partial \theta_z} &= \frac{\alpha_u(p_x \frac{\partial r_{11}}{\partial \theta_z} + p_y \frac{\partial r_{12}}{\partial \theta_z} + p_z \frac{\partial r_{13}}{\partial \theta_z})}{d}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial e_y}{\partial t_x} &= 0 \\
\frac{\partial e_y}{\partial t_y} &= \frac{\alpha_v}{d} \\
\frac{\partial e_y}{\partial t_z} &= \frac{v_0}{d} - \frac{n_y}{d^2} \\
\frac{\partial e_y}{\partial \theta_x} &= \frac{p_y(\alpha_v \frac{\partial r_{22}}{\partial \theta_x} + v_0 \frac{\partial r_{32}}{\partial \theta_x}) + p_z(\alpha_v \frac{\partial r_{23}}{\partial \theta_x} + v_0 \frac{\partial r_{33}}{\partial \theta_x})}{d} - \frac{n_y}{d^2}\left(p_y \frac{\partial r_{32}}{\theta_x} + p_z \frac{\partial r_{33}}{\theta_x}\right) \\
\frac{\partial e_y}{\partial \theta_y} &= \frac{p_x(\alpha_v \frac{\partial r_{21}}{\partial \theta_y} + v_0 \frac{\partial r_{31}}{\partial \theta_y}) + p_y(\alpha_v \frac{\partial r_{22}}{\partial \theta_y} + v_0 \frac{\partial r_{32}}{\partial \theta_y}) + p_z(\alpha_v \frac{\partial r_{23}}{\partial \theta_y} + v_0 \frac{\partial r_{33}}{\partial \theta_y})}{d} - \\
&\quad \frac{n_y}{d^2}\left(p_x \frac{\partial r_{31}}{\theta_y} + p_y \frac{\partial r_{32}}{\theta_y} + p_z \frac{\partial r_{33}}{\theta_y}\right) \\
\frac{\partial e_y}{\partial \theta_z} &= \frac{\alpha_v(p_x \frac{\partial r_{21}}{\partial \theta_z} + p_y \frac{\partial r_{22}}{\partial \theta_z} + p_z \frac{\partial r_{23}}{\partial \theta_z})}{d}
\end{aligned}
$$

The partial derivatives of the entries of the rotation matrix R are given in Table 2.

$f(\theta_x, \theta_y, \theta_z)$	$\frac{\partial f}{\partial \theta_x}$	$\frac{\partial f}{\partial \theta_y}$	$\frac{\partial f}{\partial \theta_z}$
$r11 = c_z c_y$	0	$-c_z s_y$	$-s_z c_y$
$r12 = -s_z c_x + c_z s_y s_x$	$s_z s_x + c_z s_y c_x$	$c_z c_y s_x$	$-c_z c_x - s_z s_y s_x$
$r13 = s_z s_x + c_z s_y c_x$	$s_z c_x - c_z s_y s_x$	$c_z c_y c_x$	$c_z s_x - s_z s_y c_x$
$r21 = s_z c_y$	0	$-s_z s_y$	$c_z c_y$
$r22 = c_z c_x + s_z s_y s_x$	$-c_z s_x + s_z s_y c_x$	$s_z c_y s_x$	$-s_z c_x + c_z s_y s_x$
$r23 = -c_z s_x + s_z s_y c_x$	$-c_z c_x - s_z s_y s_x$	$s_z c_y c_x$	$s_z s_x + c_z s_y c_x$
$r31 = -s_y$	0	$-c_y$	0
$r32 = c_y s_x$	$c_y c_x$	$-s_y s_x$	0
$r33 = c_y c_x$	$-c_y s_x$	$-s_y c_x$	0

Table 2: Entries of the Euler angle rotation matrix R and their partial derivatives.

A.3.2 Rotation as a unit quaternion

Given a rigid transformation represented by a translation vector $\mathbf{t}$ and a unit quaternion $\mathbf{q}$ we have:

$$\mathbf{q} = [s, q_x, q_y, q_z]^T = [s, \mathbf{u}]^T$$
$$s = \sqrt{1 - q_x^2 - q_y^2 - q_z^2}$$

The point in the camera coordinate system is:

$$\mathbf{t} + \mathbf{p} + 2[s(\mathbf{u} \times \mathbf{p}) + \mathbf{u} \times (\mathbf{u} \times \mathbf{p})]$$

The image point $\widehat{\mathbf{p}}$ is now obtained by applying the projection matrix K (Equation 3):

$$\begin{aligned} \widehat{p_x} &= \frac{\alpha_u(t_x+p_x+2(sc_x+q_yc_z-q_zc_y))+u_0(t_z+p_z+2sc_z+2(q_xc_y-q_yc_x))}{t_z+p_z+2(sc_z+q_xc_y-q_yc_x)} \\ \widehat{p_y} &= \frac{\alpha_v(t_y+p_y+2(sc_y+q_zc_x-q_xc_z))+v_0(t_z+p_z+2(sc_z+q_xc_y-q_yc_x))}{t_z+p_z+2(sc_z+q_xc_y-q_yc_x)} \end{aligned} \tag{10}$$

where:

$$\begin{aligned} c_x &\equiv (\mathbf{u} \times \mathbf{p})_x = q_y p_z - q_z p_y \\ c_y &\equiv (\mathbf{u} \times \mathbf{p})_y = q_z p_x - q_x p_z \\ c_z &\equiv (\mathbf{u} \times \mathbf{p})_z = q_x p_y - q_y p_x \end{aligned}$$

To apply the Levenberg-Marquardt optimization we differentiate the error terms $e_x(\mathbf{v})$ and $e_y(\mathbf{v})$ with respect to $\mathbf{v} = [t_x, t_y, t_z, q_x, q_y, q_z]$. Note that this approach implicitly ensures that the computed quaternion is a unit quaternion (i.e. a valid rotation).

Defining:

$$\begin{aligned} d &\equiv t_z + p_z + 2(sc_z + q_xc_y - q_yc_x) \\ n_x &\equiv \alpha_u(t_x + p_x + 2(sc_x + q_yc_z - q_zc_y)) + u_0(t_z + p_z + 2(sc_z + q_xc_y - q_yc_x)) \\ n_y &\equiv \alpha_v(t_y + p_y + 2(sc_y + q_zc_x - q_xc_z)) + v_0(t_z + p_z + 2(sc_z + q_xc_y - q_yc_x)) \end{aligned}$$

We differentiate the error terms e_x, e_y given in Equation 8 with respect to $\mathbf{v} = [t_x, t_y, t_z, q_x, q_y, q_z]$:

$$
\begin{aligned}
\frac{\partial e_x}{\partial t_x} &= \frac{\alpha_u}{d} \\
\frac{\partial e_x}{\partial t_y} &= 0 \\
\frac{\partial e_x}{\partial t_z} &= \frac{u_0}{d} - \frac{n_x}{d^2} \\
\frac{\partial e_x}{\partial q_x} &= 2\left(\frac{\alpha_u(q_y p_y + q_z p_z - \frac{q_x c_x}{s}) + u_0(s p_y + q_z p_x - 2q_x p_z - \frac{q_x c_z}{s})}{d} - \right. \\
&\quad \left. \frac{n_x}{d^2}\left(s p_y + q_z p_x - 2q_x p_z - \frac{q_x c_z}{s}\right)\right) \\
\frac{\partial e_x}{\partial q_y} &= 2\left(\frac{\alpha_u(s p_z + q_x p_y - 2q_y p_x - \frac{q_y c_x}{s}) + u_0(q_z p_y - s p_x - 2q_y p_z - \frac{q_y c_z}{s})}{d} - \right. \\
&\quad \left. \frac{n_x}{d^2}\left(q_z p_y - s p_x - 2q_y p_z - \frac{q_y c_z}{s}\right)\right) \\
\frac{\partial e_x}{\partial q_z} &= 2\left(\frac{\alpha_u(q_x p_z - s p_y - 2q_z p_x - \frac{q_z c_x}{s}) + u_0(q_x p_x + q_y p_y - \frac{q_z c_z}{s})}{d} - \right. \\
&\quad \left. \frac{n_x}{d^2}\left(q_x p_x + q_y p_y - \frac{q_z c_z}{s}\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial e_y}{\partial t_x} &= 0 \\
\frac{\partial e_y}{\partial t_y} &= \frac{\alpha_v}{d} \\
\frac{\partial e_y}{\partial t_z} &= \frac{v_0}{d} - \frac{n_y}{d^2} \\
\frac{\partial e_y}{\partial q_x} &= 2\left(\frac{\alpha_v(q_y p_x - s p_z - 2q_x p_y - \frac{q_x c_y}{s}) + v_0(s p_y + q_z p_x - 2q_x p_z - \frac{q_x c_z}{s})}{d} - \right. \\
&\quad \left. \frac{n_y}{d^2}\left(s p_y + q_z p_x - 2q_x p_z - \frac{q_x c_z}{s}\right)\right) \\
\frac{\partial e_y}{\partial q_y} &= 2\left(\frac{\alpha_v(q_z p_z + q_x p_x - \frac{q_y c_y}{s}) + v_0(q_z p_y - s p_x - 2q_y p_z - \frac{q_y c_z}{s})}{d} - \right. \\
&\quad \left. \frac{n_y}{d^2}\left(q_z p_y - s p_x - 2q_y p_z - \frac{q_y c_z}{s}\right)\right) \\
\frac{\partial e_y}{\partial q_z} &= 2\left(\frac{\alpha_v(q_y p_z + s p_x - 2q_z p_y - \frac{q_z c_y}{s}) + v_0(q_x p_x + q_y p_y - \frac{q_z c_z}{s})}{d} - \right. \\
&\quad \left. \frac{n_y}{d^2}\left(q_y p_y + q_x p_x - \frac{q_z c_z}{s}\right)\right)
\end{aligned}
$$

A.4 Genetic algorithm

Genetic algorithms are a heuristic, evolution inspired, technique for solving optimization problems. Using a collection of solutions (population) from which better and better solutions can be derived via selective breeding and mutation.

In contrast with classical non-linear optimization methods genetic algorithms do not require the optimized functions to be differentiable and that they do not require initialization close to the minimum in order to converge to it, coarse initialization may suffice.

We now give a brief description of the "simple" genetic algorithm as described in [11] and implemented in [31].

We start with a few definitions:

- genome/chromosome - An encoding of the problem parameters.
- fitness function - Quality of specific genome.

- crossover operator - Given two genomes, create two new genomes by randomly choosing a point along them and swapping all the data which appears after this point:

$$\begin{array}{ccccc|cc} (& 1 & 0 & 1 & | & 1 & 1 &) \\ (& 1 & 1 & 1 & | & 0 & 0 &) \end{array} \quad \Rightarrow \quad \begin{array}{ccccccc} (& 1 & 0 & 1 & 0 & 0 &) \\ (& 1 & 1 & 1 & 1 & 1 &) \end{array}$$

  The crossover operator may be extended to multiple point crossover and in the extreme to uniform crossover.

- mutation operator - Given a single genome, randomly change the value of a single entry, gene, in it or change the value of the whole genome.

- allele - Values a specific gene admits (discrete or continuous).

- elitism - The best genome in the population is copied into the next population as is.

The "simple" genetic algorithm proceeds as follows:

1. Create an initial population.

2. Randomly select pairs of genes from the population according to their fitness function where each gene's probability for being chosen is proportional to its fitness function (roulette wheel selection).

3. Randomly choose a crossover point and apply the crossover operator.

4. Mutate each of the new genes (values are limited by the allele settings).

5. If the algorithm allows elitism then replace the worst genome in the new population by the best one from the previous population.

6. If stopping criteria is not satisfied return to step 2, otherwise stop.

There are many problem specific stopping criteria which can be used. The most common ones are: pre-specified number of generations, population variability is below a certain threshold (all chromosomes have similar values) and, best chromosome's fitness hasn't changed for the last k generations.

Hybrid algorithms which employ both genetic algorithms and non-linear optimization also fit into the genetic algorithm framework. The role of the genetic algorithm in a hybrid algorithm is to increase the probability of converging to the *global* minimum. In a hybrid algorithm the mutation operation may perform non-linear optimization, parallel hill climbing, instead of random mutations. Another option is to perform non-linear optimization with the parameter values obtained using a genetic algorithm.

We performed pose estimation using the "simple" genetic algorithm. The algorithm uses a uniform crossover, random mutation, elitism, and minimizes the sum of squares function given in Equation 7. Each genome represents the rigid transformation using

translation and Euler angles. Using a rotation represented as a unit quaternion is possible but does not fit gracefully into the natural framework of the algorithm, requiring renormalization after every operation.

References

[1] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[3] C. Brack et al. Accurate x-ray based navigation in computer-assisted orthopedic surgery. In *Proc. Computer Assisted Radiology and Surgery*, pages 716–722, 1998.

[4] R. J. Brumback. The rationales of interlocking nailing of the femur, tibia, and humerus: An overview. *Clinical Orthopaedics and Related Research*, 324:292–320, 1996.

[5] K. Cleary and C. Nguyen. State of the art in surgical robotics: clinical applications and technology challenges. *Computer Aided Surgery*, 6(6):312–328, 2001.

[6] O. Faugeras. *Three-Dimensional Computer Vision : A Geometric Viewpoint.* MIT Press, 1993.

[7] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.

[9] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, 1980.

[10] D. Forsyth, J. L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant descriptors for 3-d object recognition and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):971–991, 1991.

[11] D. E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison Wesley, 1989.

[12] R. Halíř and J. Flusser. Numerically stable direct least squares fitting of ellipses. In *Proc. 6th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG)*, 1998.

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

[14] E. J. Hazan and L. Joskowicz. Computer-assisted image-guided intramedullary nailing of femoral shaft fractures. *Techniques in Orthopaedics*, 18(2):130–140, 2003.

[15] R. Hofstetter, M. Slomczykowski, M. Sati, and L. P. Nolte. Fluoroscopy as an imaging means for computer-assisted surgical navigation. *Computer Aided Surgery*, 4(2):65–76, 1999.

[16] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.

[17] R. Jain, R. Kasturi, and B. G. Schunk. *Machine Vision*. McGraw-Hill, 1995.

[18] L. Joskowicz. Fluoroscopy-based navigation in computer-aided orthopaedic surgery. In *Proc. of the IFAC Conf. on Mechatronic Systems*. Elsevier, 2000.

[19] L. Joskowicz, C. Milgrom, M. Shoham, Z. Yaniv, and A. Simkin. A robot-assisted system for long bone intramedullary distal locking: concept and preliminary results. In *Proc. Computer Assisted Radiology and Surgery*, pages 485–491, 2003.

[20] L. Joskowicz and R. H. Taylor. Computers in imaging and guided surgery. *IEEE Computers in Science and Engineering*, 3(5):65–72, 2001.

[21] K. Kanatani and W. Liu. 3d interpretation of conics and orthogonality. *CVGIP: Image Understanding*, 58(3):286–301, 1993.

[22] C. Krettek et al. A mechanical distal aiming device for distal locking in femoral nails. *Clinical Orthopaedics*, 384:267–275, 1999.

[23] H. Livyatan, Z. Yaniv, and L. Joskowicz. Robust automatic c-arm calibration for fluoroscopy-based navigation: a practical approach. In *Proc. of Medical Image Computing and Computer Assisted Intervention*, pages 60–68, 2002.

[24] H. B. Nielsen. Damping parameter in marquardt's method. Technical Report IMM-REP-1999-05, Department of Mathematical Modelling, Technical University of Denmark, 1999.

[25] L. P. Nolte and R. Ganz, editors. *Computer Assisted Orthopaedic Surgery (CAOS)*. Hogrefe and Huber, 1999.

[26] C. A. Rothwell, A. Zisserman, C. I. Marinos, D. A. Forsyth, and J. L. Mundy. Relative motion and pose from arbitrary plane curves. *Image and Vision Computing*, 10(4):250–262, 1992.

[27] M. Shoham, M. Burman, E. Zehavi, L. Joskowicz, E. Batkilin, and Y. Kunicher. Bone-mounted miniature robot for surgical procedures: Concept and clinical applications. *IEEE Transactions on Robotics and Automation*, 19(5):893–901, 2003.

[28] S. Skjeldal and S. Backe. Interlocking medullary nails – radiation doses in distal targeting. *Archives of Orthopaedic Trauma Surgery*, 106:179–181, 1987.

[29] R. H. Taylor and D. Stoianovici. Medical robotics in computer-integrated surgery. *IEEE Transactions on Robotics and Automation*, 19(5):765–781, 2003.

[30] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[31] M. Wall. Galib: A c++ library of genetic algorithm components. http://lancet.mit.edu/ga, 1996.

[32] Z. Yaniv, L. Joskowicz, A. Simkin, et al. Fluoroscopic image processing for computer-aided orthopedic surgery. In *Proc. Medical Computing and Computer-Assisted Intervention*, pages 325–334, 1998.

[33] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.