Chapter 1

Monocular Pose Estimation of Kinematic Chains

Bodo Rosenhahn, Oliver Granert, Gerald Sommer [1]

ABSTRACT In this paper conformal geometric algebra is used to formalize an algebraic embedding for the problem of monocular pose estimation of kinematic chains. The problem is modeled on the base of several geometric constraint equations. In conformal geometric algebra the resulting equations are compact and clear. To solve the equations we linearize and iterate the equations to approximate the pose and the kinematic chain parameters.

1.1 Introduction

In this work we derive an algebraic embedding for monocular pose estimation of kinematic chains. Pose estimation itself is a basic visual task and several approaches for monocular pose estimation exist to relate the position of a 3D object to a reference camera coordinate system (eg. [8, 7]). Instead of using invariances as an explicit formulation of geometry as often has been done in projective geometry, we are using implicit formulations and use constraints to describe the pose estimation problem. The formulas in [12] produces compact constraint equations for pose estimation of rigid objects for different situations. In many approaches the rigidity of objects is assumed, but we are also interested in kinematic chains and so to estimate the locations of bit by bit rigid objects which can change internal in a known manner. Examples are tracked robot arms or human body movements.

In this paper we will use the conformal geometric algebra (ConfGA) [4] to describe scenarios for kinematic chains [2] and their coupling with the pose estimation problem. For this we will derive a suited object representation for kinematic chains and follow the idea of the twist representation [2] to approximate the movements in a linear manner as described in [3]. Then we combine them with the pose estimation algorithm [11] to gain linear equations, which converge during iteration to the unknown pose and the internal angular or distance positions of the kinematic chain objects.

The paper is organized as follows: The first section describes the pose

[1]Institut für Informatik, Preusserstr. 1-9, 24105 Kiel, Germany,
{bro,olg,gs}@ks.informatik.uni-kiel.de

estimation scenario for rigid objects and the embedding of the scenario in ConfGA. Then we will derive a kinematic chain representation in ConfGA and describe the pose estimation constraint equations for such kinematic chains. The fourth section is devoted to the experiments in simulated and real environments and the last section ends the paper with a discussion.

1.2 Pose estimation in conformal geometric algebra

In this section we will explain the pose estimation scenario for rigid objects and their geometric representation in ConfGA. This is a summary of [12].

1.2.1 The scenario of pose estimation

In the scenario of figure 1.1 we describe the following situation: We assume

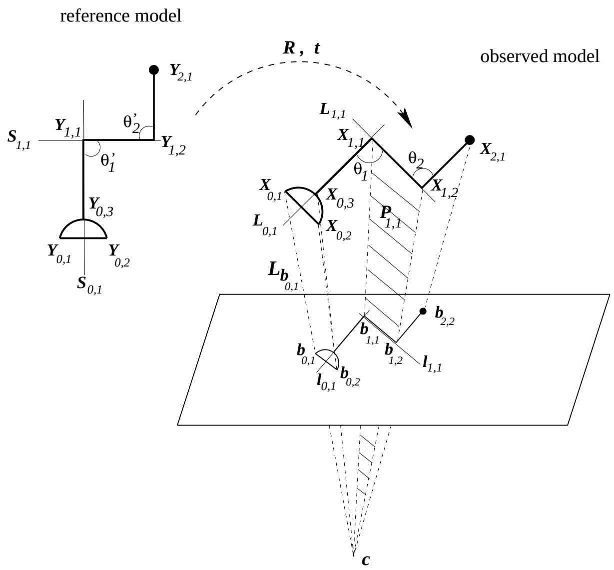

FIGURE 1.1. The scenario. The solid lines describe the assumptions: the camera model, the model of the object and the extracted lines or points on the image plane. The dashed lines describe the pose, which leads to the best fit of the object with the extracted entities.

3D points $\boldsymbol{Y}_{i,j}$, and lines $\boldsymbol{S}_{i,j}$ of an object or reference model. We denote the object features with pairs of numbers, to distinguish later between the points on the different object segments. Let us first assume the reference model as a rigid object, so that eg. the angles θ_1', θ_2' in figure 1.1 do not change during the motion. Further, we extract line subspaces $\boldsymbol{l}_{i,j}$, or points $\boldsymbol{b}_{i,j}$ in an image of a calibrated camera, whose optical centre is denoted by $\boldsymbol{c}$, and match them with the model. The aim is to find the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$, which leads to the best fit of the reference model with the actual extracted entities. To compare the image features with the object features, we interpretate the extracted image entities, resulting from the perspective projection, as a one dimensional higher entities by their back projection in the space. This idea [13] will be used to formulate the scenario as a pure kinematic problem and three different constraints can be formulated to describe the collinearity of a reference point or line to an image point or line:

1. **3D point 3D line correspondence:** A transformed point, e.g. $\boldsymbol{X}_{0,1}$, of the model point $\boldsymbol{Y}_{0,1}$ must lie on the projection ray $\boldsymbol{L}_{b_{0,1}}$, given by $\boldsymbol{c}$ and the corresponding image point $\boldsymbol{b}_{0,1}$.
2. **3D point 3D plane correspondence:** A transformed point, e.g. $\boldsymbol{X}_{1,1}$, of the model point $\boldsymbol{Y}_{1,1}$ must lie on the projection plane $\boldsymbol{P}_{1,1}$, given by $\boldsymbol{c}$ and the corresponding image line $\boldsymbol{l}_{1,1}$.
3. **3D line 3D plane correspondence:** A transformed line, e.g. $\boldsymbol{L}_{1,1}$, of the model line $\boldsymbol{S}_{1,1}$ must lie on the projection plane $\boldsymbol{P}_{1,1}$, given by $\boldsymbol{c}$ and the corresponding image line $\boldsymbol{l}_{1,1}$.

The aim in [12] is to use ConfGA to embed the scenario in a suitable algebraic language. For this the entities, the transformation of the entities and constraints for collinearity and coplanarity of involved entities are described in ConfGA. Furthermore it can be shown [13] that these constraints contain some kind of distance measure, so that they can be used as error measure to be minimized in an optimization process.

1.2.2 Introduction to conformal geometric algebra

A geometric algebra $\mathcal{G}_{p,q,r}$ is built from a vector space $\mathbb{R}^n$, endowed with the signature (p,q,r), $n=p+q+r$, by application of a geometric product. The geometric product consists of an outer ($\wedge$) and an inner ($\cdot$) product, whose roles are to increase or to decrease the order of the algebraic entities, respectively. For later use we introduce the commutator $\underline{\times}$ and anticommutator $\overline{\times}$ products, respectively for any two multivectors,

$$\boldsymbol{AB} \quad = \quad \frac{1}{2}(\boldsymbol{AB}+\boldsymbol{BA})+\frac{1}{2}(\boldsymbol{AB}-\boldsymbol{BA}) =: \boldsymbol{A}\overline{\times}\boldsymbol{B}+\boldsymbol{A}\underline{\times}\boldsymbol{B}.$$

For a discussion of these two products and their relation to the geometric, inner and outer product, see [6].

To introduce the ConfGA, we follow [4] and start with the *Minkowski plane* $\mathcal{G}_{1,1,0}$, which has an orthonormal basis $\{\mathbf{e}_+,\mathbf{e}_-\}$, defined by the properties

$$\mathbf{e}_+^2=1, \quad \mathbf{e}_-^2=-1 \text{ and } \quad \mathbf{e}_+\cdot\mathbf{e}_-=0.$$

A *Null basis* can now be introduced by the vectors

$$\mathbf{e}_0 := \tfrac{1}{2}(\mathbf{e}_- - \mathbf{e}_+) \text{ and } \quad \mathbf{e} := \mathbf{e}_- + \mathbf{e}_+.$$

Furthermore we define $\boldsymbol{E} := \mathbf{e}\wedge\mathbf{e}_0$.

In an n-dimensional vector space, the Minkowski model $\mathcal{G}_{n+1,1,0}$ will be

Constraint	Entities	ConfGA
point-line	$\underline{\boldsymbol{X}} = \boldsymbol{E} + \mathrm{e}\boldsymbol{x}$ $\underline{\boldsymbol{L}} = \boldsymbol{E}\boldsymbol{n} + \mathrm{e}\boldsymbol{M}$	$\underline{\boldsymbol{X}} \underline{\times} \underline{\boldsymbol{L}} = 0$
point-plane	$\underline{\boldsymbol{X}} = \boldsymbol{E} + \mathrm{e}\boldsymbol{x}$ $\underline{\boldsymbol{P}} = \boldsymbol{E}\boldsymbol{P} + \mathrm{e}d\boldsymbol{I}$	$\underline{\boldsymbol{X}} \underline{\times} \underline{\boldsymbol{P}} = 0$
line-plane	$\underline{\boldsymbol{L}} = \boldsymbol{E}\boldsymbol{n} + \mathrm{e}\boldsymbol{M}$ $\underline{\boldsymbol{P}} = \boldsymbol{E}\boldsymbol{P} + \mathrm{e}d\boldsymbol{I}$	$\underline{\boldsymbol{L}} \overline{\times} \underline{\boldsymbol{P}} = 0$

TABLE 1.1. The geometric constraints for pose estimation expressed in conformal geometric algebra.

used, therefore enlarging the Geometric Algebra of the n-dimensional vector space by two additional basis vectors, which define a *Null space*.

The general form of the points $\boldsymbol{x} \in \boldsymbol{\mathcal{G}}_{n,0,0}$ can be described by $\underline{\boldsymbol{x}} \in \boldsymbol{\mathcal{G}}_{n+1,1,0}$ with $\underline{\boldsymbol{x}} = \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^2\mathrm{e} + \mathrm{e}_0 =: F(\boldsymbol{x})$.
Lines can be described by the outer product of two points on the line and the point at infinity (see [5]), $\underline{\boldsymbol{L}} = \mathrm{e} \wedge \underline{\boldsymbol{a}} \wedge \underline{\boldsymbol{b}}$.
Since the outer product of three points determines a circle [4], the line can be interpreted as a circle passing through the point at infinity.

Planes can be described by the outer product of three points on the plane, and the point at infinity, $\underline{\boldsymbol{P}} = \mathrm{e} \wedge \underline{\boldsymbol{a}} \wedge \underline{\boldsymbol{b}} \wedge \underline{\boldsymbol{c}}$.
Using $\mathrm{e} \wedge \underline{\boldsymbol{a}}$ instead of $\underline{\boldsymbol{a}}$ (this is the so called *affine* representation of a point [4]), we can write the point, line and plane as

$$\begin{array}{lcll}
\underline{\boldsymbol{X}} = \mathrm{e} \wedge \underline{\boldsymbol{x}} & = & \boldsymbol{E} + \mathrm{e}\boldsymbol{x} & \\
\underline{\boldsymbol{L}} = \mathrm{e} \wedge \underline{\boldsymbol{a}} \wedge \underline{\boldsymbol{b}} & = & \boldsymbol{E}(\boldsymbol{b} - \boldsymbol{a}) + \mathrm{e}\boldsymbol{a} \wedge \boldsymbol{b} & = \boldsymbol{E}\boldsymbol{n} + \mathrm{e}\boldsymbol{M} \\
\underline{\boldsymbol{P}} = \mathrm{e} \wedge \underline{\boldsymbol{a}} \wedge \underline{\boldsymbol{b}} \wedge \underline{\boldsymbol{c}} & = & \boldsymbol{E}(\boldsymbol{b} - \boldsymbol{a}) \wedge (\boldsymbol{c} - \boldsymbol{a}) + \mathrm{e}\boldsymbol{a} \wedge \boldsymbol{b} \wedge \boldsymbol{c} & = \boldsymbol{E}\boldsymbol{P} + \mathrm{e}d\boldsymbol{I}_3.
\end{array}$$

As in other geometric algebras as well, rotations can be described by rotors $\boldsymbol{R}$. A translation can be described by a translator, $\boldsymbol{T}_{\boldsymbol{a}} = (1 + \frac{\boldsymbol{a}}{2})\mathrm{e}_0$, which is nothing more than a special rotor. Indeed, it can be shown [9] that translations, rotations, dilations and inversions can all be described by suitable rotors in ConfGA. To describe the combination of a rotation $\boldsymbol{R}$ and a translation $\boldsymbol{t}$ we denote it, according to [1], as a motor $\boldsymbol{M} = \boldsymbol{R}\boldsymbol{T}_{\boldsymbol{a}}$ which is an abbreviation of “moment and vector”.

1.2.3 Kinematic constraints in conformal geometric algebra

Now we need only formulate the constraints for collinearity and coplanarity of involved entities. Table 1.1 gives an overview of the formulations of the constraints for collinearity and coplanarity in conformal geometric algebra, which were developed and analysed in [13, 12]. These constraints contain some kind of distance measure and it can be schown [13], that the relations between the different entities are controlled by their orthogonal distance, the Hesse distance. This property leads to well conditioned equations and robustness in case of noisy data.

Combining these constraints with a rigid body motion of object points or lines, the pose estimation constraint equations reduce to setting the commutator and anticommutator products to zero [13, 12]. Thus, the constraint equations of pose estimation read

$$(\boldsymbol{M}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{M}}) \,\underline{\times}\, \underline{\boldsymbol{L}} = 0, \quad (\boldsymbol{M}\underline{\boldsymbol{X}}\widetilde{\boldsymbol{M}}) \,\underline{\times}\, \underline{\boldsymbol{P}} = 0, \quad (\boldsymbol{M}\underline{\boldsymbol{L}}\widetilde{\boldsymbol{M}}) \,\overline{\times}\, \underline{\boldsymbol{P}} = 0.$$

These compact equations subsume the pose estimation problem at hand: find the best motor $\boldsymbol{M}$ which satisfies the constraint.

1.3 Pose estimation of kinematic chains

In this section we extend the pose estimation scenario of figure 1.1 to kinematic chains. This means, eg. that the angles θ_1', θ_2' of figure 1.1 can change during the motion.

In the first subsection we describe kinematic chains in conformal geometric algebra. Then we continue with the formalization of constraints for pose estimation of kinematic chains.

1.3.1 Kinematic chains in conformal geometric algebra

So far we have parameterized the 3D pose constraint equations of a rigid object. Let be described the rigid object as a list of points. Assume that a second rigid body is attached to the first one by a joint. The joint can be formalized as an axis of rotation and/or translation in the object frame. If the joint is only dependend on a variable angle θ_i, it is called a revolute joint, and it is called a prismatic joint if the degree of freedom is only a variable length d_i [1]. The transformation of the attached points can be represented by a motor $\boldsymbol{M_1}$. For a short description of the transformations we define

$$\begin{aligned} T_0(\underline{\boldsymbol{X}}_{0,i_0}) &:= \underline{\boldsymbol{X}}_{0,i_0} \\ T_1(\underline{\boldsymbol{X}}_{1,i_1}, \boldsymbol{M}_1) &:= \boldsymbol{M}_1\underline{\boldsymbol{X}}_{1,i_1}\widetilde{\boldsymbol{M}_1}. \end{aligned}$$

This means that T_0 describes the identity for points which are not subject to internal transformations. We call them *base* points. The function T_1 formalize the transformation of an attached joint.

In the general case, a point $\underline{\boldsymbol{X}}_{n,i_n}$ of an n-th joint can be represented by a sequence of such motors $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_n$. This leads to a function T_n,

$$T_n(\underline{\boldsymbol{X}}_{n,i_n}, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_n) \quad := \quad \boldsymbol{M}_1 \ldots \boldsymbol{M}_n \underline{\boldsymbol{X}}_{n,i_n} \widetilde{\boldsymbol{M}_n} \ldots \widetilde{\boldsymbol{M}_1}.$$

An object model $\mathcal{O}$ of a kinematic chain with n segments can now be represented by a list of such functions T_i,

$$\begin{aligned} \mathcal{O} \;=\; \{ & T_0(\underline{\boldsymbol{X}}_{0,i_0}), T_1(\underline{\boldsymbol{X}}_{1,i_1}, \boldsymbol{M}_1), \ldots, T_n(\underline{\boldsymbol{X}}_{n,i_n}, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_n) \\ & |n, i_0 \ldots i_n \in \mathbb{N}\} \end{aligned}$$

Note, that the j-th joint consists of points $\underline{\boldsymbol{X}}_{j,1}, \ldots, \underline{\boldsymbol{X}}_{j,i_j}$. This numbering is also shown in figure 1.1.

1.3.2 Constraint equations for kinematic chains

In this subsection we will combine the introduced representation of kinematic chains with the pose estimation constraints, derived in section 1.2.3. This is very simple now because everything is formulated in the same algebra. The general unknown pose corresponds to a motor $\boldsymbol{M}$. For the base points $\underline{\boldsymbol{X}}_{0,i_0}$ the constraint equations reduce for a suitable projection ray $\underline{\boldsymbol{L}}$ to

$$\begin{aligned} (\boldsymbol{M}(T_0(\underline{\boldsymbol{X}}_{0,i_0}))\widetilde{\boldsymbol{M}}) \underline{\times} \underline{\boldsymbol{L}} &= 0 \\ \Leftrightarrow (\boldsymbol{M}\underline{\boldsymbol{X}}_{0,i_0}\widetilde{\boldsymbol{M}}) \underline{\times} \underline{\boldsymbol{L}} &= 0. \end{aligned}$$

The general constraint equation for a point at the j-th joint leads to

$$\begin{aligned} (\boldsymbol{M}(T_j(\underline{\boldsymbol{X}}_{j,i_j}, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_j))\widetilde{\boldsymbol{M}}) \underline{\times} \underline{\boldsymbol{L}} &= 0 \\ \Leftrightarrow (\boldsymbol{M}(\boldsymbol{M}_1 \ldots \boldsymbol{M}_j \underline{\boldsymbol{X}}_{j,i_j} \widetilde{\boldsymbol{M}_j} \ldots \widetilde{\boldsymbol{M}_1})\widetilde{\boldsymbol{M}}) \underline{\times} \underline{\boldsymbol{L}} &= 0. \end{aligned}$$

It is also simple to use extracted image lines and their reconstructed projection planes $\underline{\boldsymbol{P}}$. For such situations, the constraint equations reduce to

$$\begin{aligned} (\boldsymbol{M}(T_0(\underline{\boldsymbol{X}}_{0,i_0}))\widetilde{\boldsymbol{M}}) \underline{\times} \underline{\boldsymbol{P}} &= 0 \\ \Leftrightarrow (\boldsymbol{M}\underline{\boldsymbol{X}}_{0,i_0}\widetilde{\boldsymbol{M}}) \underline{\times} \underline{\boldsymbol{P}} &= 0, \end{aligned}$$

for the base points, and the general constraint equation for a point at the jth joint leads to

$$\begin{aligned} (\boldsymbol{M}(T_j(\underline{\boldsymbol{X}}_{j,i_j}, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_j))\widetilde{\boldsymbol{M}}) \underline{\times} \underline{\boldsymbol{P}} &= 0 \\ \Leftrightarrow (\boldsymbol{M}(\boldsymbol{M}_1 \ldots \boldsymbol{M}_j \underline{\boldsymbol{X}}_{j,i_j} \widetilde{\boldsymbol{M}_j} \ldots \widetilde{\boldsymbol{M}_1})\widetilde{\boldsymbol{M}}) \underline{\times} \underline{\boldsymbol{P}} &= 0. \end{aligned}$$

Note, that it is also possible to describe kinematic chains by lines and combine them with the LP-constraint. For this, only lines $\underline{\boldsymbol{L}}_{j,i_j}$ and projection planes $\underline{\boldsymbol{P}}_{j,i_j}$ have to be substituted and combined with the anticommutator product. Note that we always need base points for a suitable solution because it is not possible to differ between $\boldsymbol{M}$ and $\boldsymbol{M}_1$ for the first segment. This results from the geometry of the scenario and the combination of the pose estimation problem with kinematic chains.

To gain linear equations we use the exponential representation of rotors, and use the Taylor expression of fist degree for approximation. This leads to a mapping of the above mentioned global model to such one, which enables incremental changes of pose. The approximation is comparable to the *twist* description and approximation of kinematic chains, described in [3, 2]. It leads to a linear equation system, which results in a first approximation of the unknowns. Figure 1.2 visualizes such an approximation: The aim is to rotate a point $\boldsymbol{P}$ around 90 degree to a point $\boldsymbol{P'}$. The first order approximation of the rotation leads to the tangent of the circle passing

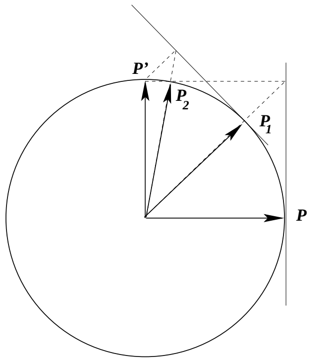

FIGURE 1.2. Principle of the convergence rate for the iteration of a point $\boldsymbol{P}$ rotated around 90 degree to a point $\boldsymbol{P'}$. $\boldsymbol{P_1}$ is the result of the first iteration and $\boldsymbol{P_2}$ is the result of the second iteration.

through $\boldsymbol{P}$. The approximation of the rotation leads to the closest point on the tangent line to $\boldsymbol{P'}$ (denoted by dashed lines), and again normalizing the point leads to $\boldsymbol{P_1}$ as the first approximation of the rotation. Now we can repeat this procedure to estimate points $\boldsymbol{P_2}, \ldots, \boldsymbol{P_n}$ which converge during the iteration to the point $\boldsymbol{P'}$. It is clear from figure 1.2, that the convergence rate of a rotation is dependend on the amount of the expected rotation. An analysis of the convergence rate for general angles is described in the next section.

1.4 Experiments

In this section we first simulate the convergence rate of a rotation during iterations. Then we test the performance of the algorithm on real images. For this experiment we use the XL-constraint, and we mark points by hand. The convergence rate of iterations for a general rotation θ is demonstrated in figure 1.3. The x-axis represents the angle θ, the y-axis shows the estimated angle $\hat{\theta}$. Four iterations are overlaid. The functions are very characteristic and it can be seen, that the contribution of the first iteration to gain 90 degree rotation is 45 degree. This is clear comparing the situation with figure 1.2.

All angles, except that of 180 degree converge during the iteration and for the most cases only a few iterations are sufficient to get a good approximation. For situations, where only small rotations are assumed, for the most cases, two or three iterations are sufficient.

The following experiments visualize the application of the pose estimation algorithm on real scenarios, see figures 1.4, 1.5 and 1.6. In the first image sequence, the object model is a door in a cupboard and both the angle of the door and the robot are changing. During these movements we extract the correspondences by hand and visualize the transformed projected object in the sequence. It is easy to see, that both unknowns, the

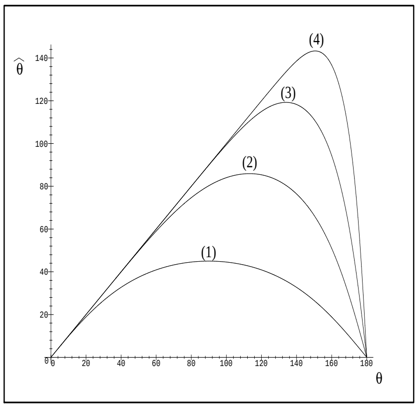

FIGURE 1.3. Convergence rate of iterations for arbitrary angles between 0 and 180 degrees. The expected angles θ are on the x-axis and the evaluated angles $\hat{\theta}$ are on the y-axis. The iterations $(1)\ldots(4)$ are overlaid.

pose of the cupboard and the angle of the door are estimated and the error is very small.

In the second image sequence, the object model is a doll and we estimate the pose, the angle of the upper arm and the forearm. Figure 1.5 visualizes the transformed projected object in the sequence. Though we only used one 3D point for each kinematic chain segment and measured the size of the doll by hand, the pose is also accurate.

In the third image sequence, we use as object model a robot arm. We estimate the pose of the robot and the angles of the kinematic chain via tracked points markers. Dependend on the position of the camera with respect to the object model and the location of the joints the estimated angles differ around 0.5 till 3 degrees to the ground truth. Figure 1.6 visualizes some results.

1.5 Discussion

This paper presents an algebraic embedding for monocular pose estimation of kinematic chains. Conformal geometric algebra is well suited to model the involved geometric scenario since both the pose estimation problem and the representation of kinematic chains are compact and easy to combine. The involved geometry is implicitly represented and described on the base of several geometric constraint equations. Any deviations from the constraints correspond to the Hesse distance of the involved geometric entities [11]. So it is possible to ensure well conditioned equations systems. The linearization and iteration of the constraint equations is easy to implement and it is shown, that only a few iterations are necessary to get a good approximation of the pose and the kinematic chain parameters.

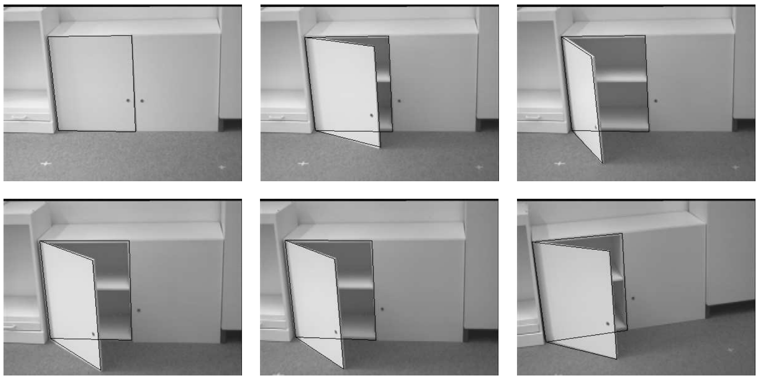

FIGURE 1.4. Images of the first real scenario. Both the pose of the cupboard and the opening angle of the door are estimated.

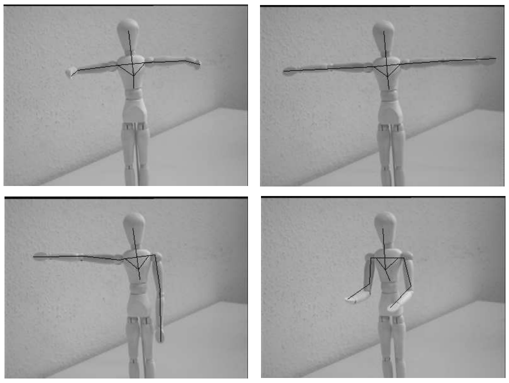

FIGURE 1.5. Images of the second real scenario. The pose of the doll and the angles of the arms are estimated.

Acknowledgements

We would like to thank Norbert Krüger, Christian Perwass and Joan Lasenby for fruitfull discussions and hints performing this work.

References

[1] Bayro-Corrochano E. and Kähler D. Kinematics of Robot Manipulators in the Motor Algebra *In Sommer G., editor, Geometric Computing with Clifford Algebra. Springer Verlag*, 2001.

[2] Murray R.M., Li Z. and Sastry S.S. A Mathematical Introduction to Robotic Manipulation. *CRC Press*, 1994.

[3] Bregler C. and Malik J. Tracking People with Twists and Exponential Maps. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, pp.8-15 1998.

[4] Li H. Generalized Homogeneous Coordinates for Computational Geometry *In Sommer G., editor, Geometric Computing with Clifford Algebra., Springer Verlag*,pp

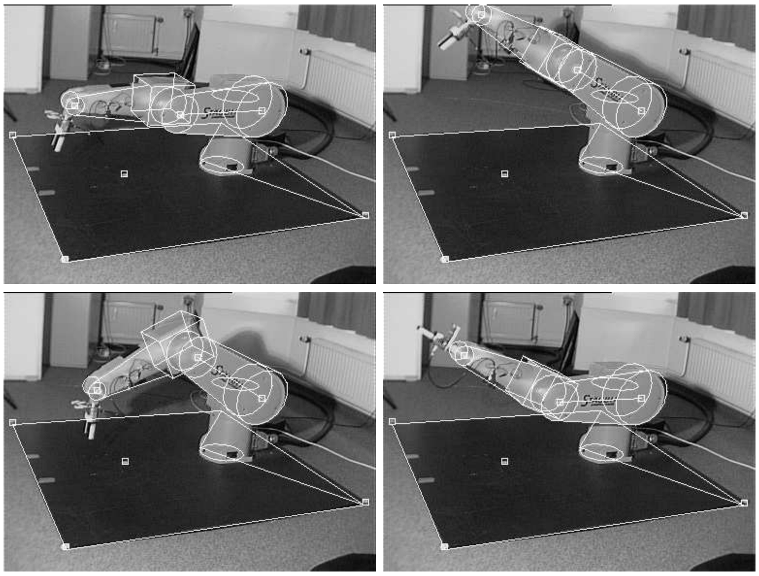

FIGURE 1.6. Images of the third real scenario. The pose of the robot and the opening angles of the arm are estimated.

27-52, 2001.

[5] Needham T. Visual Complex Analysis. *Oxford University Press*, 1997

[6] Perwass C. and Lasenby J. A novel axiomatic derivation of geometric algebra. Technical Report CUED/F - INFENG/TR.347, Cambridge University Engineering Department, 1999.

[7] Walker M.W. and Shao L. Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54,3: 358–367, 1991.

[8] Horaud R., Phong T.Q. and Tao P.D. Object pose from 2-d to 3-d point and line correspondences. *International Journal of Computer Vision*, 15: 225–243, 1995.

[9] Lasenby A. and Lasenby J. Surface Evolution and Representation using Geometric Algebra *The Mathematics of Surfaces IX* Proceedings of the ninth IMA conference on the mathematics of surfaces. Ed. Roberto Cipolla and Ralph Martin, p144-168, Springer, London, 2000.

[10] Rosenhahn B., Krüger N., Rabsch T., Sommer G. Tracking with a Novel Pose estimation algorithm. *International Workshop on Robot Vision; Auckland, New Zealand, R. Klette, G. Sommer, S. Peleg (Eds.), LNCS 1998*, Springer, Heidelberg, pp. 9-19, 2001.

[11] Rosenhahn B., Zhang Y. and Sommer G. Pose estimation in the language of kinematics. *Second International Workshop, Algebraic Frames for the Perception-Action Cycle, AFPAC 2000, LNCS 1888*, Springer, Heidelberg, pp.284-293, 2000.

[12] Rosenhahn B. and Lasenby J. Constraint Equations for 2D-3D Pose Estimation in Conformal Geometric Algebra. Technical Report CUED/F - INFENG/TR.396, Cambridge University Engineering Department, 2000.

[13] G. Sommer, B. Rosenhahn and Y. Zhang. Pose estimation using geometric constraints. *Technical Report 2003,Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik*, 2000.