Tracking with a Novel Pose Estimation Algorithm

Bodo Rosenhahn, Norbert Krüger, Torge Rabsch, Gerald Sommer

Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Preußerstrasse 1-9, 24105 Kiel, Germany
{bro,nkr,tr,gs}@ks.informatik.uni-kiel.de

Abstract. In this paper we apply a novel pose estimation algorithm to the tracking problem. We make use of error measures of the algorithm which enable us to characterize the quality of an estimated pose. The key idea of the tracking algorithm is random start local search. The principle of the heuristic relies upon a combination of iterative improvement and random sampling. While in many approaches a manually designed object representation is assumed, we overcome this condition by using accumulated object representations and combine these successfully with the tracking algorithm.

1 Introduction

In this work we apply a novel 2D-3D pose estimation algorithm [12] to the tracking problem. This algorithm shows some interesting characteristics which makes it especially useful for this purpose. Beside features such as stability in the presence of noise and online–capabilities its main advantage in the tracking context is that it can unify different kinds of correspondences within one algebraic framework.

To apply the pose estimation algorithm to the tracking problem we intend to solve two problems which were avoided in [12] but are important for further applications like robot navigation or object recognition:

1. **Correspondences:** Correspondences between model data and image data have been defined manually.
2. **Object Representation:** A manually designed representation of the object to be tracked has been presupposed.

In this paper we describe an automatic procedure to find correspondences between an object model and its image projection which makes use of features of the pose estimation algorithm [12] and of the specific tracking condition. We suppose a 3D model of the object consisting of 3D points and 3D lines and we extract lines in the image sequence by a Hough transformation combined with a new algorithm to extract lines from the Hough array. We find correspondences between 3D lines and 2D lines by a local search. The essential attribute is that a discrete local neighborhood of states is defined with respect to the current state, in this context the Hamming distance n–neighborhood [11]. Further, we allow correspondences only for entities with small distance. This assumption is justified by the specific tracking situation. The pose estimation algorithm is able to use correspondences as 3D point to 2D point, 3D point to 2D line and 3D line to 2D line to estimate the rotation and translation between two frames. In this paper only line correspondences are used. Note that this kind of correspondence allows to avoid the so called appertur problem, i.e. the impossibility to define correspondences between a point on a line in two frames.

To avoid a manually designed object representation we also applied the tracking algorithm with an accumulated object representation consisting of local 3D line segments. The object accumulation is based on a scheme which accumulates confidences for entities representing the object and which allows to extract representations in even quite complicated environments [4]. We could show, that with such a representation tracking is possible and therefore both assumptions of manual intervention in [12] can be substituted by automatic procedures.

2 Description of the Tracking

In this context tracking means to minimize a matching error by solving two problems:

1. The correspondence problem: Determine the mapping between model elements (here 3D model lines) and image features (extracted Hough lines).
2. The spatial fitting problem (pose estimation): For each correspondence determine the best parameters (here rotation R and translation t), so that the spatial fit error of the model lines to image lines is minimized.

In the following sections we describe the automatic extraction of lines (section 2.1), the pose estimation algorithm (section 2.2), the automatic finding of correspondences (section 2.3), and the accumulating of object representations (section 2.4).

2.1 Hough Transformation

To extract lines in an image we apply the well known Hough transformation [3]. The robustness of the Hough transformation can be increased by using not only information about the presence of edges but by also checking the agreement of lines and local orientation, i.e. by applying the orientation selective Hough transformation [9]. The Hough transformation results in an accumulator array (see figure 1) from which the representative lines show up as peaks. These are easily detectable for 'simple' images such as the one in figure 1 but difficult to extract in more complex situations.

To avoid the extraction of additional lines caused by locally neighbored peaks in the accumulator array (often occurring in the presence of noise in the image data) usually some kind of metric on the accumulator array is defined to allow only lines corresponding to peaks with certain distance. A problem of these methods is that important lines may have small distance in the Hough space (see e.g., narrow parallel lines in figure 2). To extract the significant lines we also use information about the areas which do support lines, i.e. we evaluate also image information. This allows us to extract lines with small distance in the accumulator array which are usually not extractable by other methods (for details see [1]).

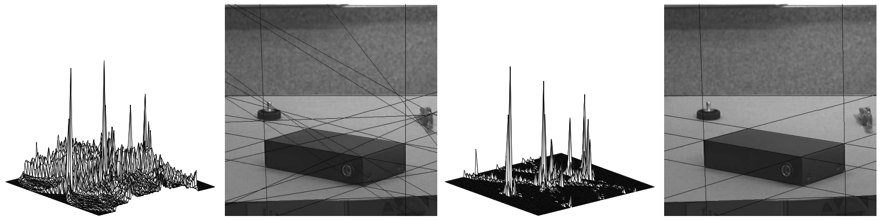

Fig. 1. Standard-Hough-transformation and Orientation selective Hough-transformation

Fig. 2. Representative Hough lines extracted by different methods

Figure 2 shows extracted Hough lines using different kind of metrics. In the left image our method has been used, in the middle image for each selected peak a neighborhood in the accumulator array is set to zero (as, e.g. in [8]), while in the right image connected areas which occur after thresholding the accumulator array are treated as one line (as e.g., in [6]). Note that the narrow parallel lines could only be extracted by our method. The procedure used in the middle image extracts the most significant lines but not the narrow parallel lines because the corresponding peaks are too close in the accumulator array. The procedure used for the right image has great difficulties with locally neighbored peaks which are above threshold.

2.2 Pose Estimation

The problem of pose estimation means to estimate the transformation (the rigid body motion) between the two coordinate frames of measured data and model data. In [12, 10] the problem of 2D-3D pose estimation is described in the algebraic language of kinematics. The key idea is that the observed 2D entities together with their corresponding 3D entities are constraint to lie on other, higher order entities which result from the perspective projection. The observed 2D entities in this context are extracted Hough lines.

To be more detailed, in the scenario of figure 3 we describe the following situation: We assume 3D points $\boldsymbol{Y_i}$, and lines $\boldsymbol{S}_i$ of an object or reference model. Further, we extract line subspaces $\boldsymbol{l}_i$ in an image of a calibrated camera and match them with the model. Three constraints can be depicted:

1. **3D point 2D point correspondence:** A transformed point, e.g. $\boldsymbol{X_1}$, of the model point $\boldsymbol{Y_1}$ must lie on the projection ray $\boldsymbol{L}_{b1}$, given by $\boldsymbol{c}$ and the corresponding image point $\boldsymbol{b_1}$.
2. **3D point 2D line correspondence:** A transformed point, e.g. $\boldsymbol{X_1}$, of the model point $\boldsymbol{Y_1}$ must lie on the projection plane $\boldsymbol{P_{12}}$, given by $\boldsymbol{c}$ and the corresponding image line $\boldsymbol{l}_1$.
3. **3D line 2D line correspondence:** A transformed line, e.g. $\boldsymbol{L}_1$, of the model line $\boldsymbol{S}_1$ must lie on the projection plane $\boldsymbol{P_{12}}$, given by $\boldsymbol{c}$ and the the corresponding image line $\boldsymbol{l}_1$.

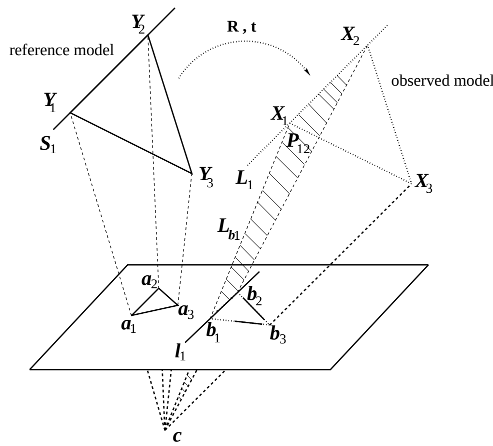

Fig. 3. The scenario. The solid lines at the left describe the assumptions: the camera model, the model of the object and the initially extracted lines on the image plane. The dashed lines at the right describe the actual pose of the model, which leads to the best fit of the object with the actual extracted lines.

The use of the motor algebra [2] allows to subsume the pose estimation problem by compact constraint equations since the entities, the transformation of the entities and the constraints can be described economically in one unifying language. Furthermore the constraint equations express a natural distance measure, in this case the Hesse distance between the entities, which is also explained in [12]. This property is important for the robustness of our algorithms since we work with digital images and noisy data. To solve these constraint equations a special extended motor Kalman filter was developed [13].

2.3 Testing of Correspondences

It is well known, that for $l = m \times n$ potential pairs, there are $S = 2^{|l|}$ correspondences. This means, the search space is in general very large and not practicable for applications. The tracking assumption allows to use local criteria like distances and angles to reduce the search space significantly, depending on the error boundaries. In this context the correspondence space for m model lines

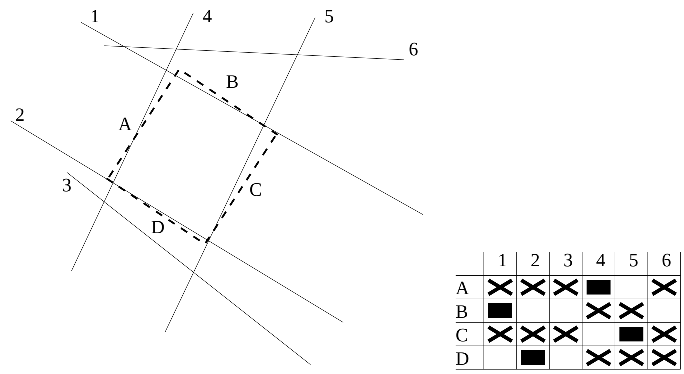

Fig. 4. Match example for a rectangle. The model lines are labeled with letters and the extracted image lines are labeled with numbers. The table indicates the correspondence space with the allowed possibilities (white/black), the impossible matches (cross) and the current match (black).

and n image lines is represented by a $m \times n$ fit-matrix. In this matrix flags represent the needed information for a match, mismatch or potential match, figure 4 shows an example. In this example the model lines are labeled with letters and the extracted image lines are labeled with numbers. The table indicates the correspondence space with the allowed matches (white/black), the impossible matches (cross) and the current match (black). See also [11] for further information.

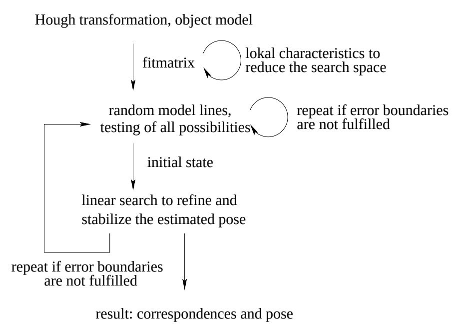

Fig. 5. A scheme of the tracking algorithm.

Random start local search [11] is the basis for our algorithm, which is summarized in figure 5. The principle of the heuristic relies upon a combination of iterative improvement and random sampling. Iterative improvement refers to a repeated generate-and-test principle by which the algorithm moves from an initial state to its local optimum. So the algorithm consists of two main steps: First find an initial state for a minimum of correspondences and then refine the result by the other correspondences. For the first step we choose five random

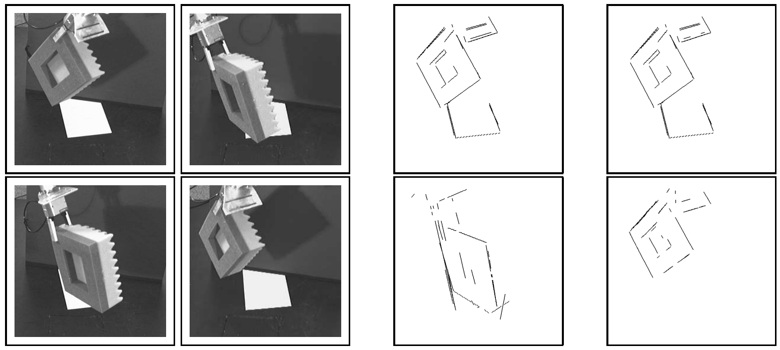

Fig. 6. Accumulation of an object representation (first and fifth iteration). The robot has physical control over the object. Line segments corresponding to the background vanish after a few iterations. Left: the stereo images of left and right camera. Middle: Representation extracted from one stereo image pair. Right: Accumulated representation.

model lines and try every combination of the object lines to the allowed image lines to estimate an optimal pose and use the error function to characterize the quality of the pose. This is possible since the error measure corresponds directly to the Hesse distance and leads to a suitable error measure. Once the initial pose is estimated, in the second step an additional model line will be tried to match an allowed image line to stabilize and refine the result. Note, that this part of the algorithm is linear, since the use of the Kalman filter leads to recognizable peaks for the detection of mismatches [14]. So the first assumption of [12], i.e. the knowledge of the correspondences can be solved by the algorithm, which is summarized in figure 5.

2.4 Object accumulation

The second assumption, i.e., a manually designed object model can be avoided by applying the methods described above with a model extracted from a stereo image sequence. The key idea of the algorithm (described more precisely in [4] and [5]) is to accumulate evidences for entities used to represent an object over time. In our case the object was manipulated by a robot (see figure 6). This allows us to solve the correspondence problem during accumulation since the knowledge of the parameters of motion could be used in the accumulation scheme. Here the entities used to represent an object are local 3D line segments. However, the accumulation scheme can be applied for a wide range of visual entities. After forty iterations the object model was good enough to be applied in our tracking algorithm.

3 Experiments

In our first experimental scenario we use a manually designed model of a house for tracking. Figure 7 shows some results of the sequence with the superimposed model of the house. The slight displacements between the model and the house

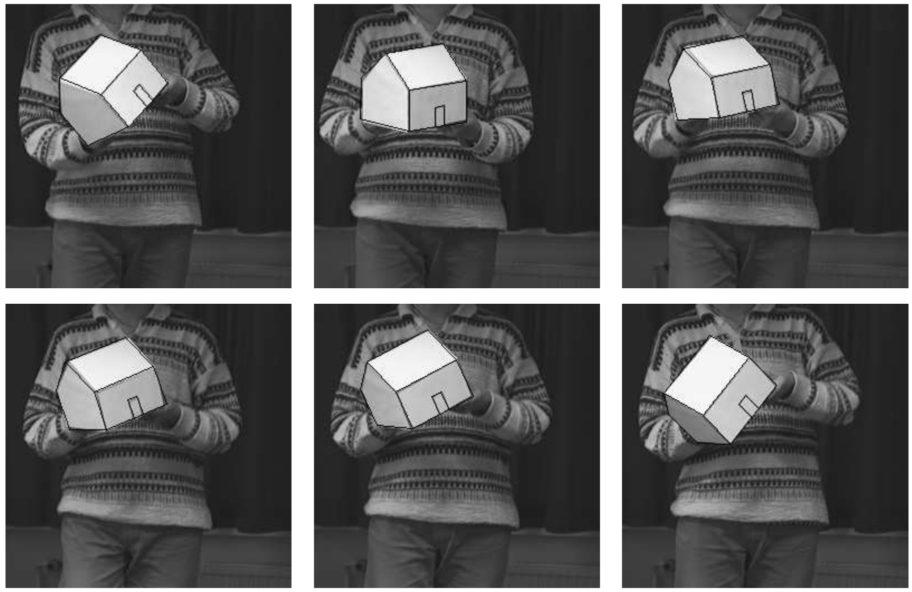

Fig. 7. Tracking with a manually designed object representation.

on the image in some of the frames emerge from calibration errors, extraction errors and match errors.

In our second experimental scenario we accumulate an object representation of a model house by the algorithm described in section 2.4. Our accumulated object model consists of 130 line segments.[1] Though the accumulated representation also consists of noisy line segments, which do not belong to the house, the algorithm is able to estimate the transformations, which are necessary to get a good fit of the object model with the image lines. Since our algorithm is also able to neglect object lines, our algorithm is able to deal with hidden or not extracted object features in the image, or noisy line segments of the object model. Some results of the required and estimated movements are visualized in figure 8. The performance of our algorithm is not optimized yet and the main steps, the Hough transformation and the testing of correspondences are not in real time. The Hough transformation itself needs about two seconds, and the testing of the correspondences needs about 5 seconds to 15 seconds in artificial designed objects and 3 to 5 minutes with the accumulated object (because of its 130 line segments). But still the algorithm is heuristic and we also had cases where it

[1] Our object representation consists of a large number of satistically very dependent entities. For matching it would be advantageous if these entities become connected by some kind of grouping process to achieve a representation with a smaller set of more complex features to speed up matching. The formalization of such grouping processes is part of our research.

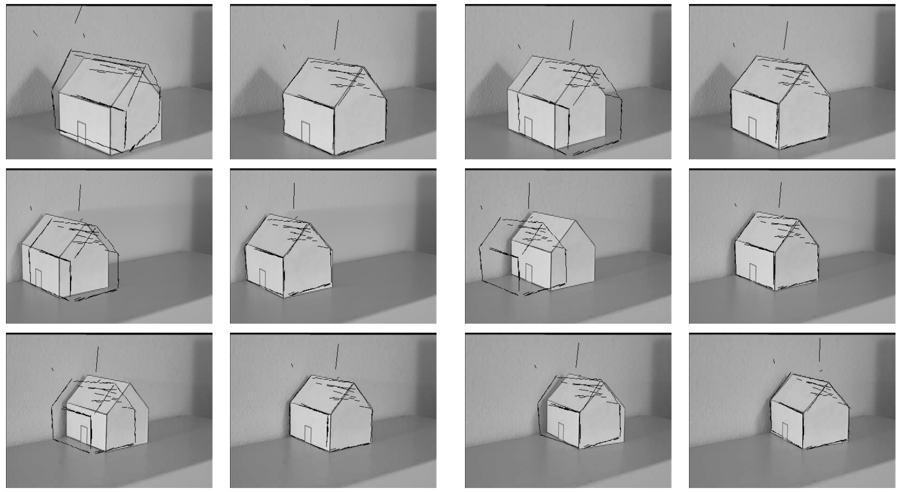

Fig. 8. Tracking with an accumulated object model. In this sequence we show the results before tracking and after tracking for each image to visualize the movements.

never converged. The time performance is also dependend on the parameters of the tracking assumptions and the parameters of the Hough transformation.

4 Conclusion and Outlook

We applied the novel pose estimation algorithm described in [12] to the tracking problem. For tracking we could automatically find correspondences between model data and Hough lines by a local search algorithm. Furthermore, we could demonstrate that tracking is even possible with an accumulated object representation.

In this paper we only used 2D line to 3D line correspondences. However, with a more elaborated object representation consisting of point features (such as corners) as well as line features, other kind of correspondences could be applied for tracking as well. The possibility to deal with these different entities within one framework as in the pose estimation algorithm in [12] would be an interesting extension of the tracking algorithm introduced here.

In this paper tracking and accumulation are distinct competences, for object accumulation it was necessary to have physical control over the object by a robot to solve the correspondence problem. With the tracking algorithm introduced here we aim to replace the need of physical control. The pose estimation gives us the parameters of object motion which are needed in our accumulation scheme and which were granted by the knowledge of the motor commands of the robot. Therefore, by combining tracking and accumulation we might achieve learning while doing object tracking.

All algorithms introduced here were implemented in the C++–software library KiViGraP [7] which allows us to combine competences as the one introduced in this paper into one system. In [5] a framework of such a system is discussed in which basic competences can be combined to more complex behavior patterns.

Acknowledgment

We would like to thank Kord Ehmcke and Oliver Granert for performing the experiments necessary to accumulate the object representation applied here. Furthermore we thank Yiwen Zhang for his implementation of the Kalman filter within KiViGraP.

References

1. M. Ackermann. Akkumulieren von Objektrepräsentationen im Wahrnehmungs–Handlungs Zyklus. *Diplomarbeit, Lehrstuhl für Kognitive Systeme der Christian-Albrechts-Universität zu Kiel*, 2000.
2. E. Bayro-Corrochano. The geometry and algebra of kinematics. *In Sommer G., editor, Geometric Computing with Clifford Algebra. Springer Verlag*, 2000.
3. P.V.C. Hough. Methods and means for recognizing complex patterns. *U.S. Patent 3,069,654, Dec. 18*, 1962.
4. Norbert Krüger, Marcus Ackermann, and Gerald Sommer. Accumulation of object representations utilizing interaction of robot action and perception. *DAGM 2000*,pp.365-372, 2000.
5. Norbert Krüger, Daniel Wendorff, and Gerald Sommer. Two models of a vision–based robotic system: Visual haptic attention and accumulation of object representations. *submitted to Robot Vision 2001.*
6. S. Kunze. Ein Hand-Auge-System zur visuell basierten Lokalisierung und Identifikation von Objekten. *Diplomarbeit, Lehrstuhl für Kognitive Systeme der Christian-Albrechts-Universität zu Kiel*, 2000.
7. KiViGraP (Homepage of the Kieler Vision and Grasping Project). http://www.ks.informatik.uni-kiel.de/~kivi/kivi.html.
8. J. Pauli. Geometric/photometric consensus and regular shape quasi-invariants for object localization and boundary extraction. *Technical Report 9805, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik*, 1998.
9. J. Princen, J. Illingworth, and J. Kittler. An optimizing line finder using a Hough transform algorithm. *Computer Vision, Graphics, and Image Processing*, 52:57–77, 1990.
10. B. Rosenhahn, Y. Zhang, and G. Sommer. Pose estimation in the language of kinematics. *Second International Workshop, Algebraic Frames for the Perception-Action Cycle, AFPAC 2000, LNCS 1888*, pp.284-293, 2000.
11. J.R. Beveridge. Local search algorithms for geometric object recognition: Optimal correspondence and pose. *Technical Report CS 93–5, University of Massachusetts*, 1993.
12. G. Sommer, B. Rosenhahn, and Y. Zang. Pose estimation using geometric constraints. *Technical Report 2003,Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik*, 2000.
13. Y. Zhang, B. Rosenhahn, and G. Sommer. Extended Kalman filter design for motion estimation by point and line observations. *Second International Workshop, Algebraic Frames for the Perception-Action Cycle, AFPAC 2000, LNCS 1888*,pp.339-348, 2000.
14. Y. Zhang and G. Sommer. 3D rigid object tracking in computer vision. *Presented in Kiel, Oberseminar , July 2000, Work in preparation.*