Robust Pose Estimation with 3D Textured Models

Juergen Gall, Bodo Rosenhahn, and Hans-Peter Seidel

Max-Planck Institute for Computer Science
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
{jgall, rosenhahn, hpseidel}@mpi-sb.mpg.de

Abstract. Estimating the pose of a rigid body means to determine the rigid body motion in the 3D space from 2D images. For this purpose, it is reasonable to make use of existing knowledge of the object. Our approach exploits the 3D shape and the texture of the tracked object in form of a 3D textured model to establish 3D-2D correspondences for pose estimation. While the surface of the 3D free-form model is matched to the contour extracted by segmentation, additional reliable correspondences are obtained by matching local descriptors of interest points between the textured model and the images. The fusion of these complementary features provides a robust pose estimation. Moreover, the initial pose is automatically detected and the pose is predicted for each frame. Using the predicted pose as shape prior makes the contour extraction less sensitive. The performance of our method is demonstrated by stereo tracking experiments.

1 Introduction

This paper addresses the task of estimating the pose of a rigid body in the 3D space from images captured by multiple calibrated cameras. For solving this problem it is a natural approach to exploit the available information on the object as far as possible. In [1] the knowledge of the 3D shape was integrated in a contour based 3D tracker. Knowing the 3D model, the estimating process relies on correspondences between some 2D features in the images and their counterparts on the 3D model. Our approach extends the work by incorporating also the texture of the object. The additional information allows to extract more reliable correspondences that makes the estimation more robust.

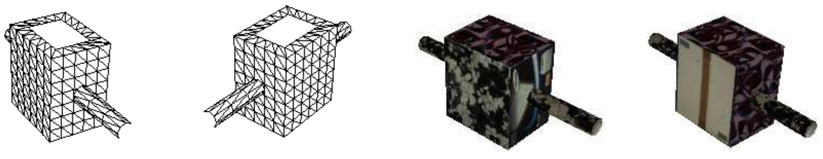

Fig. 1. 3D mesh and rendered textured model used for tracking.

There are numerous features that have been used for establishing correspondences, e.g., matching lines [2], blocks [3], local descriptors [4], and free-form contours [5].

They all work well under some conditions, however, none of them can handle general situations. The most approaches assume that the corresponding image features are visible during the whole sequence. They either completely fail when the number of features is very low caused, for example, by occlusion or they reinitialize the pose after some frames when enough features are again detected [6]. Whereas the contour extraction as described in [1] is robust to occlusions. However, the contour does not provide enough information for smooth and convex objects to estimate the pose uniquely. Furthermore, the contour extraction is only suitable for movements that are slow enough such that the segmentation does not get stuck in a local optimum. Hence, more than one feature is needed for robust tracking.

Combining the object contour with the optical flow between successive frames has been proposed in [7]. Although it performs well, it assumes that the initial pose is known and cannot recover from a significant error. Furthermore, the optical flow is easily distracted by other objects moving in front of the observed object. Our work instead combines the object contour with image features between a frame and a 3D textured model projected onto the image plane. We assume that the textured model of the object is available where the lightning conditions for capturing the texture are allowed to differ from the conditions during tracking, i.e., the model construction is independent of the tracking sequence.

Since lightning conditions between the object and its textured model are inhomogeneous and the object is transformed by a rigid body motion (RBM), we use local descriptors that provide robust matching under changes in viewpoint and illumination. A comparison of local descriptors [8] revealed that SIFT [9], PCA-SIFT [10], and GLOH [8] perform best. The descriptors build a distinctive representation of a so-called keypoint in an image from a patch of pixels in its neighborhood. The keypoints are localized by an interest point detector. We use the detector proposed by Lowe [11] based on local 3D extrema in the scale-space pyramid built with difference-of-Gaussian filters. It has the advantage that it runs faster than other detectors [12], e.g., like the slower Harris-Affine detector [13]. The DoG representation, however, is not affine invariant. Hence, we cannot use GLOH that requires an affine-invariant detector. Therefore, we used PCA-SIFT that reduces the dimension of the descriptor by principal component analysis. This speeds up the matching process and produces less outliers than SIFT but also less correspondences.

In the next section, we give an overview of the whole pose estimation process that will be discussed in detail in the following sections. Experiments in Section 5 with a 3D textured model as shown in Fig. 1 demonstrate the performance of the proposed technique. A brief discussion is given at the end.

2 Overview

Our approach for pose estimation is illustrated by the flow chart in Fig. 2. Knowing the pose of the object for frame $t - 1$, we generate a 3D textured model in the same world coordinate system used for the calibration of the cameras, see Section 4.1. Rendered images of the model are obtained by projecting the model onto the image plane according to the calibration matrix for each camera.

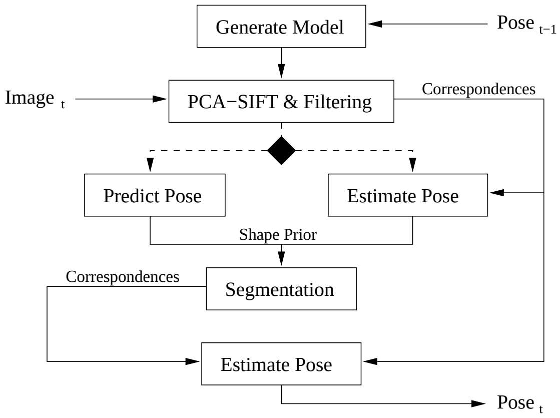

Fig. 2. Correspondences extracted by PCA-SIFT and correspondences between the contour of the projected 3D model and the contour obtained by segmentation are used for pose estimation. If not enough keypoints are detected by PCA-SIFT, an autoregression is performed to predict the pose for the next frame.

In a second step, the PCA-SIFT [10] features are extracted from the rendered images and from the new images of frame t. The features are used for establishing correspondences between the 3D model and the 2D images for each view as described in Section 4.2. In [6] and [14], RANSAC is used to estimate the pose from the matches that include outliers. RANSAC, however, is not suitable for integrating correspondences from the contour and cannot handle inaccuracy of the keypoint localizations, e.g., arising from texture registration. Therefore, we use a least-squares approach as used in [5], see Section 3. If not enough correspondences are extracted by PCA-SIFT, the pose is predicted by autoregression as discussed in Section 4.3.

The next step consists of extracting the contour by a variational model for level set based image segmentation incorporating color and texture [15] where the predicted pose is used as shape prior [1], see Section 4.4. New correspondences between the 3D model and the 2D image are then established by matching the extracted contour with the projected contour of the model via an iterated closest point algorithm [16]. Finally, the correspondences obtained from PCA-SIFT and from the segmentation are used for estimating the pose in frame t.

3 Pose Estimation

For pose estimation we assume that correspondences between the 3D model (X_i) and a 2D image (x_i) are already extracted and write each correspondence as pair (X_i, x_i) of homogeneous coordinates. In order to estimate the 3D rigid body motion M that fits best the correspondences, M is represented as exponential of a twist [17]

$$\theta\hat{\xi} = \theta \begin{pmatrix} \hat{\omega} & v \\ 0 & 0 \end{pmatrix}, \qquad \hat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \qquad \|\omega\|_2 = 1, \tag{1}$$

i.e., $M = \exp(\theta\hat{\xi})$. A twist with varying $\theta \in \mathbb{R}$ describes a screw motion in $\mathbb{R}^3$ where θ corresponds to the rotation velocity and pitch. The function $\exp(\theta\hat{\xi})$ can be efficiently computed by the Rodriguez formula [17] and linearized by $\exp(\theta\hat{\xi}) = \sum_{k=0}^{\infty}((\theta\hat{\xi})^k/k!) \approx I + \hat{\xi}$, where I denotes the identity matrix.

Each image point x_i defines a projection ray that can be represented as Plücker line [17] determined by a unique vector n_i and a moment m_i such that $x \times n_i - m_i = 0$ for all x on the 3D line. Furthermore, $\|x \times n_i - m_i\|_2$ is the norm of the perpendicular error vector between the line and a point $x \in \mathbb{R}^3$. Hence, the pose estimation consists of finding a twist such that the squared error for $(\exp(\theta\hat{\xi})X_i)_{3\times 1}$ is minimal for all pairs, where $(\cdot)_{3\times 1}$ denotes the transformation from homogeneous coordinates back to non-homogeneous coordinates. Using the linearization, we obtain for each correspondence the constraint equation

$$(\exp(\theta\hat{\xi})X_i)_{3\times 1} \times n_i - m_i = 0 \qquad (2)$$

which can be rearranged into the form $A\xi = b$. The resulting overdetermined linear system is solved by standard methods like the Householder algorithm. From the resulting twist ξ, the RBM M_1 is computed and applied to all X_i. The pose estimation is iterated until the motion converges. After n iterations, usually 3-5 are sufficient, the concatenated rigid body transformation $M = M_n \ldots M_2 M_1$ is the solution for the pose estimation. In a multi-view setting as in our experiments, the correspondences for each camera are added to one linear system and solved simultaneously. Our implementation takes about 4ms for 200 correspondences.

4 Correspondences

4.1 Textured Model

We assume that a 3D model including textures is already constructed independently of the tracking sequences, i.e., we do not require that the textures are extracted from the tracking sequences. Hence, the modelling process is done only once and the model can be reused for any sequence provided that the texture remains unchanged. In order to render the 3D model in the same coordinate system as used for camera calibration, the calibration matrices are converted to the modelview and projection matrix representation of OpenGL. Since OpenGL cannot handle lens distortions directly, the image sequences are undistorted beforehand. However, the step could also be efficiently included by a look-up table. In a preprocessing step, PCA-SIFT is trained for the object by building the patch eigenspace from the object textures. Moreover, we render some initial views of the 3D model by rotating and store the extracted keypoints, strictly speaking the PCA-SIFT descriptors of the keypoints, with the corresponding RBM. From the data, our system automatically detects the pose in the first frame.

4.2 Matching

After the 3D model is rendered and projected onto the image plane for each camera view, the keypoints are extracted by PCA-SIFT. The keypoints are also extracted from

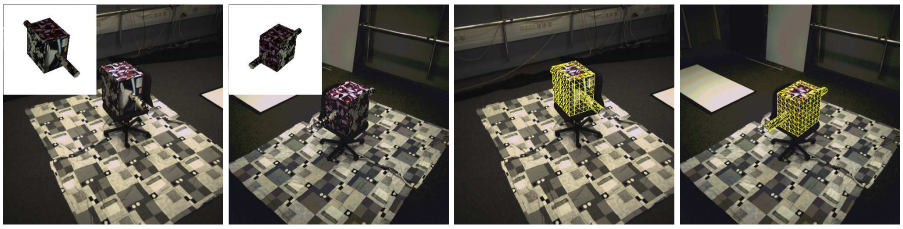

Fig. 3. Initialization. **Left:** Both camera views of the first frame. Best initial view for initialization is shown in top left corner. **Right:** Estimated pose after initialization.

the captured images. The effort is reduced by bounding cubes for each component of the 3D model. Projecting the corners of the cubes provides a 2D bounding box for each image. Since we track an object, we can assume that the object is near the bounding box except for the first frame. Hence, the detector is only performed on a subimage. 2D-2D correspondences are then established by nearest neighbor distance ratio matching [8], where we use as additional constraint that two different located points cannot correspond to points with the same position. Since the set of correspondences contains outliers, the rudest mismatches are removed by discarding correspondences with an Euclidean distance that exceeds the average by a multiple.

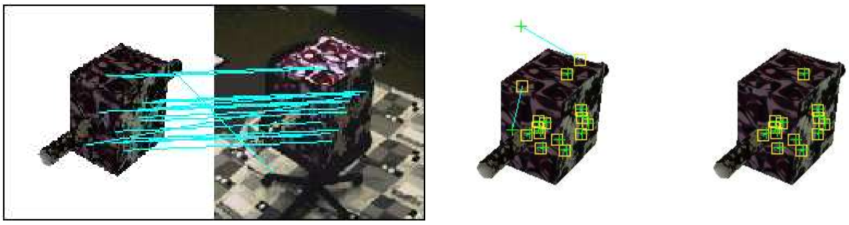

Fig. 4. Left: Correspondences between projected model and image. **Center:** Displaying the points of the projected model *(yellow squares)* corresponding to points in the image *(green crosses)*. Two outliers are in the set of correspondences. **Right:** After filtering only the outliers are removed.

The 3D coordinate X of a 2D point x in the projected image plane of the model is obtained as following: Each 2D point is inside or on the border of a projected triangle of the 3D mesh with vertices v_1, v_2, and v_3. The point can be expressed by barycentric coordinates, i.e., $x = \sum_i \alpha_i \, v_i$. Assuming an affine transformation, the 3D point is given by $X = \sum_i \alpha_i \, V_i$. The corresponding triangle for a point can be efficiently determined by a look-up table containing the color index and vertices for each triangle. After that the pose is estimated from the resulting 2D-3D correspondences. In a second filtering process, the new 3D coordinates from the estimated pose are projected back and the last outliers are removed by thresholding according to the Euclidean distance between the 2D correspondences and the reprojected counterparts.

During initialization, the keypoints from the images are matched with the keypoints extracted from the initial views beforehand. According to the number of matches, a best initial view is selected and the pose is estimated from the obtained correspondences.

4.3 Prediction

The logarithm of a RBM: In [17] a constructive way is given to compute the twist which generates a given RBM: Let $R \in SO(3)$ be a rotation matrix and $t \in \mathbb{R}^3$ a translation vector for the RBM. For the case $R = I$, the twist is given by

$$\hat{\xi} = \begin{pmatrix} 0 & \frac{t}{\|t\|} \\ 0 & 0 \end{pmatrix}, \theta = \|t\|_2. \tag{3}$$

For the other cases, the motion velocity θ and the rotation axis ω is given by

$$\theta = \cos^{-1}\left(\frac{trace(R) - 1}{2}\right), \omega = \frac{1}{2\sin(\theta)} \begin{pmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{pmatrix}. \tag{4}$$

To obtain v, the matrix

$$A = (I - \exp(\theta\hat{\omega}))\hat{\omega} + \omega\omega^T\theta, \tag{5}$$

obtained from the Rodriguez formula needs to be inverted and multiplied with the translation vector t, i.e., $v = A^{-1}t$. This follows from the fact, that the two matrices which comprise A have mutually orthogonal null spaces when $\theta \neq 0$. Hence, $Av = 0 \Leftrightarrow v = 0$. We call the transformation from $SE(3)$ to $se(3)$ the logarithm, $\log(M)$.

The adjoint transformation: It is not trivial to derive a formula for the velocity of a rigid body whose motion is given by $g(t)$, a curve parameterized by time t in $SE(3)$, since $SE(3)$ is not Euclidean. In particular, $\dot{g} \notin SE(3)$ and $\dot{g} \notin se(3)$. But by representing a rigid body motion as a screw action, the spatial velocity can be represented by the twist of the screw, see [17] for details. This allows for motion interpolation, damping and prediction.

Later we will take the motion history P_i of the last N frames into account. For a suited prediction we use a set of twists $\xi_i = \log(P_i P_{i-1}^{-1})$ representing the relative motions. To generate a suited *average* rigid body motion we make use of the adjoint transformation to represent a screw motion with respect to another coordinate system: If $\xi \in se(3)$ is a twist given in a coordinate frame A, then for any $G \in SE(3)$ which transforms a coordinate frame A to B, is $G\hat{\xi}G^{-1}$ a twist with the twist coordinates given in the coordinate frame B, see [17] for details. The mapping $\hat{\xi} \longmapsto G\hat{\xi}G^{-1}$ is called the *adjoint transformation* associated with G.

Given a set of world positions and orientations P_i the twists ξ_i can be used to express the motion as local transformation in the current coordinate system M_1: Let $\xi_1 = \log(P_2 P_1^{-1})$ be the twist representing the relative motion from P_1 to P_2. This

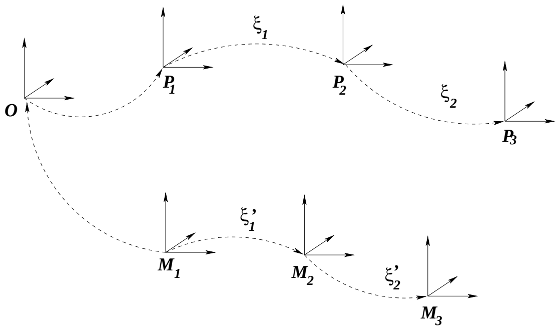

Fig. 5. Transformation of rigid body motions from prior data P_i in a current world coordinate system M_i. A proper scaling of the twists results in a proper damping.

transformation can be expressed as local transformation in the current coordinate system M_1 by the adjoint transformation associated with $G = M_1 P_1^{-1}$. The new twist is then given by $\hat{\xi}_1' = G\hat{\xi}_1 G^{-1}$. The advantage of the twist representation is now that the twists can be scaled by a factor $0 \leq \lambda_i \leq 1$ to damp the local rigid body motion, i.e., $\hat{\xi}_1' = G\lambda_1\hat{\xi}_1 G^{-1}$.

The average RBM from N given local rigid body motions can then be written as consecutive evaluation of such local rigid body motions scaled with $\lambda_i = 1/N$.

4.4 Segmentation

The images are segmented by a level set based method incorporating color and texture [15]. It splits the image domain Ω^i of each view into object region Ω_1^i and background region Ω_2^i by level set functions $\Phi^i : \Omega^i \to \mathbb{R}$, such that $\Phi^i(x) > 0$ if $x \in \Omega_1^i$ and $\Phi^i(x) < 0$ if $x \in \Omega_2^i$. The contour of an object is thus represented by the zero-level line. The approach described in [1] uses a variational model that integrates the contour of a prior pose $\Phi_0^i(\widehat{x})$ for each view $1 \leq i \leq r$ as shape prior. It minimizes the energy functional $E(\widehat{x}, \Phi^1, \ldots, \Phi^r) = \sum_{i=1}^r E(\widehat{x}, \Phi^i)$ where

$$E(\widehat{x}, \Phi^i) = -\int_{\Omega^i} H(\Phi^i) \ln p_1^i + (1 - H(\Phi^i)) \ln p_2^i \, dx \\ + \nu \int_{\Omega^i} \left|\nabla H(\Phi^i)\right| \, dx + \lambda \int_{\Omega^i} \left(\Phi^i - \Phi_0^i(\widehat{x})\right)^2 \, dx \tag{6}$$

and H is a regularized version of the step function.

Minimizing the first term corresponds to maximizing the a-posteriori probability of all pixel assignments given the probability densities p_1^i and p_2^i of Ω_1^i and Ω_2^i, respectively. These densities are modeled by Gaussian densities whose parameters are estimated from the previous level set function. The second term minimizes the length of the contour and smoothes the resulting contour. The last one penalizes the discrepancy to the shape prior that is obtained by projection of the predicted pose. The relative influence of the three terms is controlled by the constant weighting parameters $\nu = 0.5$ and $\lambda = 0.06$.

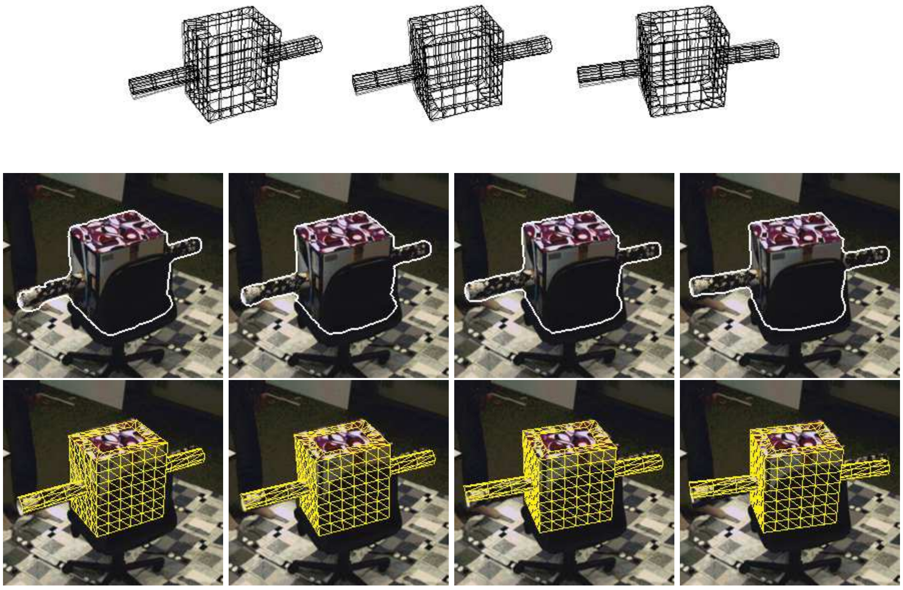

Fig. 6. 4 successive frames of a rotation sequence (only one view is shown). **Top row:** Pose is predicted by autoregression for lack of PCA-SIFT matches. *Black:* Predicted pose. *Gray:* Previous pose. **Middle row:** Contour extracted by segmentation. **Bottom row:** Estimated pose.

After segmentation, the 3D-2D correspondences for each view are given by the projected vertices of the 3D mesh that are part of the model contour and their closest points of the extracted contour determined by an iterated closest point algorithm [16].

4.5 Fusion of correspondences

Although it has been shown that the segmentation as previously described is quite robust to clutter, shadows, reflections, and noise [1], a good shape prior is essential for tracking since both matching between the contours and the segmentation itself is prone to local optima. The predicted pose by an autoregression usually provides a better shape prior than the estimated pose in the previous frame. In situations, however, where the object region and the background region are difficult to distinguish, the error of the segmentation and the error of the prediction are accumulating after some time. The shortcoming is compensated by PCA-SIFT, but it is also clear that usually not enough keypoints are available in each frame. Hence, the correspondences from contour matching and from descriptor matching are added to one linear system for the pose estimation. Since the contour provides more correspondences, the Equations (2) for the correspondences from PCA-SIFT are weighted by $\sharp Corrs_{Contour}/5$.

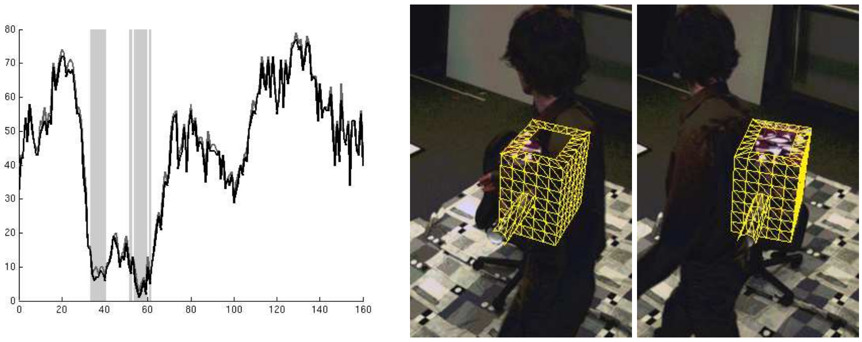

Fig. 7. Rotation sequence with a moving person. **Left:** Number of matches from PCA-SIFT *(dark gray)*. After filtering the number of matches is only slightly reduced *(black)*. When the number is below a threshold, the pose is predicted by an autoregression *(gray bars)*. **Right:** The rotating box is occluded by a moving person.

5 Experiments

For evaluating the performance of our approach, we used the 3D textured model as shown in Fig. 1. The textures were captured under different lightning conditions from the conditions for the image sequences that were recorded by two calibrated cameras. Although the size of the images is 502×502, the object is only about 100×100. The initial position was automatically detected for each sequence as shown in Fig. 3.

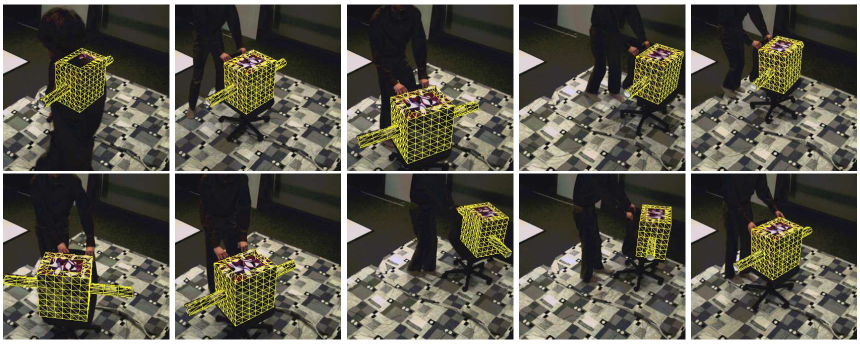

Fig. 8. Pose estimates for 10 of 570 frames. The sequence contains several difficulties for tracking: a rich textured and non-static background, shadows, occlusions, and other moving objects. Only one camera view is shown.

The tracked object is partially covered with two dissimilar customary fabrics and the printed side reflects the light. It is placed on a chair that occludes the back of the object. The background is rich textured and non-static. Shadows, dark patterns on the texture and the black chair make contour extraction difficult even for the human eye.

Furthermore, a person moves and occludes the object. These conditions make great demands on the method for pose estimation.

In the first sequence, the chair with the object rotates clockwise. When the back of the chair occludes the object, there are not enough distinctive interest points for pose estimation. Therefore, the pose is predicted by an autoregression for the next frame as shown in Fig. 6. Due to the shape prior, the segmentation is robust to the occlusion such that the estimates are still accurate. The number of matches from PCA-SIFT with respect to time is plotted in Fig. 7. During the sequence, the object rotates counterclockwise while the person orbits the object clockwise. As we can see from the diagram, PCA-SIFT produces only few outliers that are removed after the filtering. The gray bars in the diagram indicate the frames where an autoregression was performed. Since the number of matches range from 1 to 77, it is clear that an approach based only on the descriptors would fail in this situation.

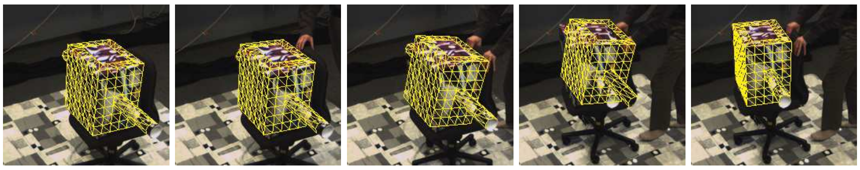

Fig. 9. Comparison with a contour-based method. **From left to right:** Pose estimates for frames 5, 50, 90, 110. **Rightmost:** Result of our method at frame 110.

Pose estimates for a third sequence including rotations and translations of the object are shown in Fig. 8. When only the contour is used, the pose estimation is erroneous since both segmentation and contour matching are distracted by local optima, see Fig. 9. For comparison, the result of our method is also given.

Finally, we simulated disturbances of the sequence in order to obtain a quantative error analysis. Since the object is placed on the chair, the y-coordinate of the pose is approximately constant. During the sequence, however, the object shifts slightly on the chair. The peak at frame 527 in the diagram of Fig. 10 is caused by a relocation of the object. For one sequence, we added Gaussian noise with standard deviation 35 to each color channel of a pixel. Another sequence was disturbed by 80 teapots that were rendered in the 3D space of the tracked object. The teapots drop from the sky where the start positions, material properties, and velocities are random. Regarding the result for the undistorted sequence as some kind of ground truth, the diagram in Fig. 10 shows the robustness of our approach. While an autoregression was performed only twice for the unmodified sequence and the average number of filtered matches per frame from PCA-SIFT was 50.9, the numbers fell down to 27.9 and 13.1 for the teapots sequence with 132 predictions and the noisy sequence with 361 predictions.

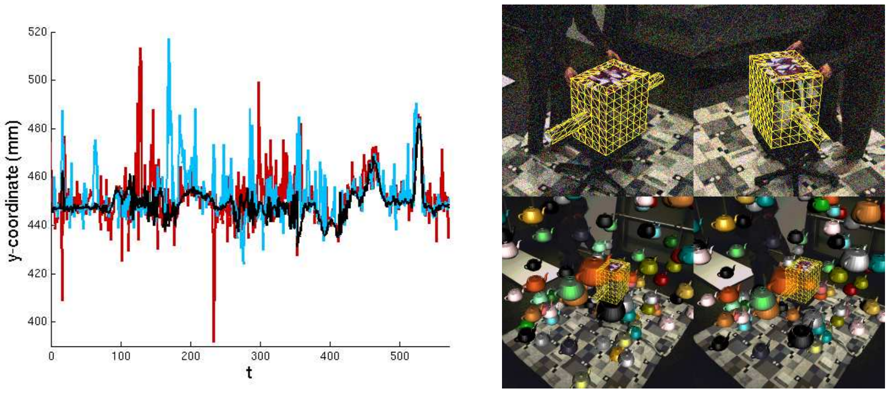

Fig. 10. Left: Quantative error analysis for a sequence with disturbances. *Black:* Undisturbed sequence. *Red:* Gaussian noise with standard deviation 35. *Blue:* 80 teapots dropping from the sky with random start position, material properties, and velocity. **Right:** *Top:* Stereo frame 527 of the noisy sequence (image details). *Bottom:* Two successive frames of the teapot sequence.

6 Conclusions

In this work, we have suggested a textured model based method for 3D pose estimation. It fuses two different features for matching, namely contour and local descriptors, where the influence of the features is automatically adapted during tracking. The initial pose is identified without supervision. In our experiments, we have demonstrated that our approach overcomes the drawbacks of the single features and that it can be applied to quite general situations. In the case of a homogeneous object without distinctive keypoints, our approach operates as a pure contour-based method. Furthermore, we have provided visual and quantative results showing that our approach is able to deal with a rich textured and non-static background and multiple moving objects. Moreover, it is robust to shadows, occlusions, and noise. Although our experiments considered only rigid bodies with a simple geometric surface, our method works with any kind of free-form objects. The pose estimation can be straightforward extended to articulated objects [18]. This will be done in future.

Acknowledgments

Our research is funded by the Max-Planck Center for visual computing and communication. We thank Yan Ke for providing the PCA-SIFT source code.

References

1. Rosenhahn, B., Brox, T., Weickert, J.: Three-dimensional shape knowledge for joint image segmentation and pose tracking. Int. J. of Computer Vision (2006)
2. David, P., DeMenthon, D., Duraiswami, R., Samet, H.: Simultaneous pose and correspondence determination using line feature. In: Int. Conf. of Computer Vision. (2003) 424–431

3. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3d tracking using online and offline information. IEEE Trans. on Pattern Analysis and Machine Intelligence **26**(10) (2004) 1391–1391
4. Allezard, N., Dhome, M., Jurie, F.: Recognition of 3d textured objects by mixing view-based and model-based representations. Int. Conf. on Pattern Recognition **01** (2000) 960–963
5. Rosenhahn, B., Perwass, C., Sommer, G.: Pose estimation of free-form contours. Int. J. of Computer Vision **62**(3) (2005) 267–289
6. Lepetit, V., Pilet, J., Fua, P.: Point matching as a classification problem for fast and robust object pose estimation. In: Conf. on Computer Vision and Pattern Recognition. Volume 2. (2004) 244–250
7. Brox T., Rosenhahn B., C.D., H.-P., S.: High accuracy optical flow serves 3-d pose tracking: Exploiting contour and flow based constraints. In Leonarids A., B.H., A., P., eds.: European Conf. on Computer Vision. Volume 3952 of LNCS., Springer (2006) 98–111
8. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Conf. on Computer Vision and Pattern Recognition **02** (2003) 257–263
9. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. of Computer Vision **60**(2) (2004) 91–110
10. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. In: IEEE Conf. on Computer Vision and Pattern Recognition. Volume 2. (2004) 506–513
11. Lowe, D.: Object recognition from local scale-invariant features. In: Int. Conf. on Computer Vision. (1999) 1150–1157
12. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. of Computer Vision **60**(1) (2004) 63–86
13. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: European Conf. on Computer Vision. (2002) 128–142
14. Brown, M., Lowe, D.: Invariant features from interest point groups. In: British Machine Vision Conf. (2002) 656–665
15. Brox, T., Rousson, M., Deriche, R., Weickert, J.: Unsupervised segmentation incorporating colour, texture, and motion. In Petkov, N., Westenberg, M.A., eds.: Computer Analysis of Images and Patterns. Volume 2756 of LNCS., Springer (2003) 353–360
16. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. Int. J. of Computer Vision (1994)
17. Murray, R., Li, Z., Sastry, S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton, FL (1994)
18. Rosenhahn, B., Brox, T., Smith, D., Gurney, J., Klette, R.: A system for marker-less human motion estimation. Künstliche Intelligenz **1** (2006) 45–51