High Accuracy Optical Flow Serves 3-D Pose Tracking: Exploiting Contour and Flow Based Constraints*

Thomas Brox[1], Bodo Rosenhahn[2], Daniel Cremers[1], and Hans-Peter Seidel[2]

[1] CVPR Group, Department of Computer Science, University of Bonn,
Römerstr. 164, 53113 Bonn, Germany
{brox, dcremers}@cs.uni-bonn.de
[2] Max Planck Center for Visual Computing and Communication,
D-66123 Saarbrücken, Germany
rosenhahn@mpi-sb.mpg.de

Abstract. Tracking the 3-D pose of an object needs correspondences between 2-D features in the image and their 3-D counterparts in the object model. A large variety of such features has been suggested in the literature. All of them have drawbacks in one situation or the other since their extraction in the image and/or the matching is prone to errors. In this paper, we propose to use two complementary types of features for pose tracking, such that one type makes up for the shortcomings of the other. Aside from the object contour, which is matched to a free-form object surface, we suggest to employ the optic flow in order to compute additional point correspondences. Optic flow estimation is a mature research field with sophisticated algorithms available. Using here a high quality method ensures a reliable matching. In our experiments we demonstrate the performance of our method and in particular the improvements due to the optic flow.

1 Introduction

To determine the 3-D pose of objects in a scene is an important task in computer vision. In this paper, we focus on the task of pose tracking, i.e., we assume the pose of the object is approximately known at the first frame of an image sequence. For not loosing this pose information over time, we seek to capture the exact 3-D object motion from one frame to the next, given an a-priori 3-D object model. The estimated motion thereby has to fit the 3-D model to some 2-D image data in the new frame. We assume rigid objects, though the concept can also be extended to more general objects modelled as kinematic chains [3, 10, 1]. So our goal is to determine 6 motion parameters, 3 for the object's rotation and 3 for its translation in space.

For estimating these parameters, one has to match 3-D features of the object model to their 2-D counterparts in the image. There are many possibilities which features to match, ranging from line matching [8] and block matching [27], up to local descriptors like SIFT [16] and free-form contour matching [23]. All of these features have their specific shortcomings. Either they are not appropriate for general objects, like lines, or they are difficult to match, consequently producing false matches that disturb the pose

* We gratefully acknowledge funding by the German Research Foundation (DFG) and the Max Planck Center for Visual Computing and Communication.

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part II, LNCS 3952, pp. 98–111, 2006.

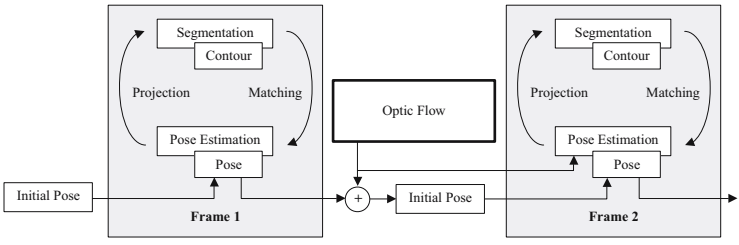

Fig. 1. Illustration of the pose tracking system. Given an initial pose, segmentation and contour based pose estimation are iterated to successively improve the extracted contour and the pose. Between frames, the optic flow helps to improve the initial pose. Furthermore, it supplements additional point correspondences for pose estimation.

estimation. The appropriateness of a feature for matching depends on the situation. In case of textured objects with many distinctive blobs, block matching and SIFT work pretty well. However, such methods may fail to match homogeneous objects with few distinctive features. Further on, block matching is only suited for translational motion and has well known problems in scenes with, e.g., rotating objects. In such cases, contour matching may work much better, as the contour is adaptive to the object's shape. However, the silhouette of very smooth and convex objects does not carry much information, and further point matches from inside the object region can be necessary to ensure unique solutions. Moreover, contour extraction and matching are susceptible to local optima.

To overcome these limitations of individual features, we propose to combine two complementary types of features. On one hand, we match the object contour extracted from the image to the object surface. This method works well for all rigid objects if two requirements are satisfied: 1) the silhouette contains enough information to provide a unique pose estimate, 2) the motion of the object from one image to the next is small enough to ensure that the contour extraction and matching do not run into a local optimum.

Additionally to the correspondences from the silhouette, we propose to add matches obtained from the optic flow, i.e., correspondences of 2-D points in successive images. If the pose in the first image is known, which is the case for pose tracking, the optic flow allows for constructing 2D-3D correspondences in the second image. Since the optic flow based features provide correspondences for points from the *interior* of the object region, they are complementary to the silhouette features and may provide uniqueness of solutions precisely in cases where the silhouettes are not sufficiently descriptive. This aims at the first shortcoming of contour based matching. To address the second shortcoming, we employ the high-end optic flow estimation method introduced in [4], which can deal with rather large displacements. With the flow based pose estimation predicting the pose in the next image, we enable a contour matching that avoids local minima and can handle much larger motion.

In experiments, we verified that the integration of these two complementary features yields a very general pose tracking approach that can deal with all kinds of rigid objects, large object motions, background clutter, light changes, noise, textured and non-textured objects, as well as partial occlusions. The information from multiple cameras can be

used and it does not matter whether the object or the camera are moving. Moreover, the interlaced contour-surface matching ensures that errors from the optic flow do not accumulate, so even after many frames the method yields precise pose estimates.

Paper organization. The next section explains the pose estimation method assuming that 2D-3D point correspondences are given. In Section 3 we then show how such correspondences can be obtained, once by matching the contour to the object surface, and once by employing 2-D correspondences obtained from the optic flow. In our experiments we demonstrate the generality of the method and show in particular the improvements due to the optic flow. Section 5 finally provides a brief summary.

Related work. There exists a wide variety of pose estimation algorithms differing by the used object or image features, the camera geometry, single or multi-view geometry, and numerical estimation procedures. For an overview see [11, 22]. The first point based techniques were studied in the 80's and 90's, and pioneering work was done by Lowe [15] and Grimson [12]. A projective formulation of Lowe's work can be found in [2]. The use of 3-D Plücker lines was investigated in [25]. Point matching by means of the optic flow has been investigated, e.g., in [14] and [3], where optic flow correspondences are used in a point-based approach with a scaled orthographic camera model. In [9] the linearized optic flow constraint is integrated into a deformable model for estimating the object motion. Block matching approaches are related to optic flow based methods, though the matching is often restricted to a few interest points. Combinations of optic flow or block matching with edge maps has been presented in [17, 26]. Recently, more enhanced local descriptors have been suggested to deal with the shortcomings of block matching. A performance evaluation can be found in [18].

2 Pose Estimation

This section describes the core algorithm for point based 2D-3D pose estimation. We assume a set of corresponding points (X_i, x_i), with 4-D (homogeneous) model points X_i and 3-D (homogeneous) image points x_i. Each image point is reconstructed to a Plücker line $L_i = (n_i, m_i)$, with a unit direction n_i, and moment m_i [19]. The 3-D rigid motion we estimate is represented in exponential form

$$\boldsymbol{M} = \exp(\theta\hat{\xi}) = \exp\begin{pmatrix} \hat{\omega} & \boldsymbol{v} \\ 0_{3\times 1} & 0 \end{pmatrix} \tag{1}$$

where $\theta\hat{\xi}$ is the matrix representation of a twist $\xi = (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3) \in se(3) = \{(\boldsymbol{v}, \hat{\omega}) | \boldsymbol{v} \in \mathbb{R}^3, \hat{\omega} \in so(3)\}$, with $so(3) = \{\hat{\omega} \in \mathbb{R}^{3\times 3} | \hat{\omega} = -\hat{\omega}^T\}$. In fact, $\boldsymbol{M}$ is an element of the one-parametric Lie group $SE(3)$, known as the group of direct affine isometries. A main result of Lie theory is, that to each Lie group there exists a Lie algebra, which can be found in its tangential space, by derivation and evaluation at its origin; see [19] for more details. The corresponding Lie algebra to $SE(3)$ is denoted as $se(3)$. A twist contains six parameters and can be scaled to $\theta\xi$ with a unit vector ω. The parameter $\theta \in \mathbb{R}$ corresponds to the motion velocity (i.e., the rotation velocity and pitch). Variation of θ corresponds to a screw motion around an axis in space. To reconstruct a group action $\boldsymbol{M} \in SE(3)$ from a given twist, the exponential function $\exp(\theta\hat{\xi}) = \boldsymbol{M} \in SE(3)$ must be computed. It can be calculated efficiently by using the Rodriguez formula [19]. For pose estimation we combine the

reconstructed Plücker lines with the screw representation for rigid motions and apply a gradient descent method: incidence of the transformed 3-D point X_i with the 3-D ray $L_i = (n_i, m_i)$ can be expressed as

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

Indeed, X_i is a homogeneous 4-D vector, and after multiplication with the 4×4 matrix $\exp(\theta\hat{\xi})$ we neglect the homogeneous component (which is 1) to evaluate the cross product with n_i. Note that this constraint equation expresses the perpendicular error vector between the Plücker line and the 3-D point. The aim is to minimize this spatial error. To this end, we linearize the equation by using $\exp(\theta\hat{\xi}) = \sum_{k=0}^{\infty} \frac{(\theta\hat{\xi})^k}{k!} \approx \boldsymbol{I} + \theta\hat{\xi}$, with $\boldsymbol{I}$ as identity matrix. This results in

$$((\boldsymbol{I} + \theta\hat{\xi})X_i)_{3\times 1} \times n_i - m_i = 0 \tag{3}$$

which can be rearranged into an equation of the form

$$\boldsymbol{A}\xi = \boldsymbol{b}. \tag{4}$$

Collecting a set of such equations (each is of rank two) leads to an over-determined linear system of equations in ξ. From the twist ξ one can reconstruct the group action $\boldsymbol{M}^1$. It is then applied to X_i which results in $X_i^1 = \boldsymbol{M}X_i$ as the result after the first iteration. The pose estimation is now repeated until the motion converges. For n iterations we get $\boldsymbol{M} = \boldsymbol{M}^n \dots \boldsymbol{M}^1$ as pose of X_i to x_i. Usually 3-5 iterations are sufficient for an accurate pose.

In this setting, the extension to multiple views is straightforward: we assume N images which are calibrated with respect to the same world coordinate system and are triggered. For each camera the system matrices $\boldsymbol{A}_1 \dots \boldsymbol{A}_N$ and solution vectors $\boldsymbol{b}_1 \dots \boldsymbol{b}_N$ are generated. The equations are now bundled in one system $\boldsymbol{A} = (\boldsymbol{A}_1, \dots, \boldsymbol{A}_N)^T$ and $\boldsymbol{b} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_N)^T$. Since they are generated for the same unknowns ξ, they can be solved simultaneously, i.e., the spatial errors of all involved cameras are minimized.

In conclusion, given the projection matrices of the cameras and a set of 2D-3D correspondences, pose estimation comes down to solve an overdetermined linear system of equations, which takes typically 4ms for 200 point correspondences. The remaining problem of pose estimation is hence how to compute reliable point correspondences, i.e., how to match features visible in the image to features of the object model.

3 Feature Matching

The following two sections are concerned with the computation of contour based and optic flow based point correspondences. For both the contour extraction [7, 6] and the optic flow estimation [4], rather sophisticated methods are employed. We focus on describing only the models of these techniques and how they affect the pose estimation. Implementation details, such as numerical schemes, can be found in the abovementioned papers and the references therein.

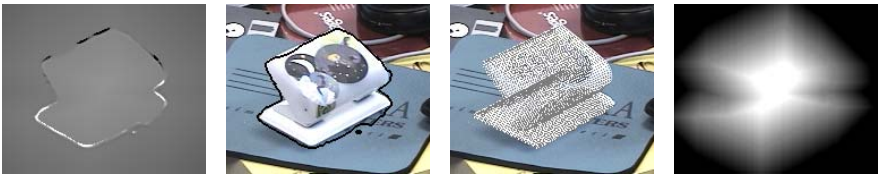

Fig. 2. Illustration of contour representations by means of level set functions. **From Left to Right:** (a) Level set function Φ. (b) Contour represented by the zero-level line of Φ. (c) Object projected to the image, given the current pose. (d) Shape prior Φ_0 derived from the object silhouette.

3.1 Contour-Surface Matching

Contour extraction. The computation of contour based correspondences is according to our prior work in [5]. It builds upon contour extraction by means of region based level set segmentation [7, 21]. In such methods, one provides an initial contour and evolves this contour for that it becomes optimal with regard to some energy model. This energy functional reads in our case:

$$\begin{aligned} E(\Phi) = & \underbrace{-\int_\Omega \big(H(\Phi(x)) \log p_1(F(x)) + (1 - H(\Phi(x))) \log p_2(F(x)\big)\, dx}_{\text{Region Statistics}} \\ & \underbrace{+\nu \int_\Omega |\nabla H(\Phi(x))|\, dx}_{\text{Contour Smoothness}} + \underbrace{\lambda \int_\Omega (\Phi(x) - \Phi_0(x))^2\, dx}_{\text{Shape}} \quad \rightarrow \min . \end{aligned} \tag{5}$$

Hereby the level set function Φ represents the contour by its zero-level line [20, 7]; see Fig. 2a,b for an illustration. $H(\Phi)$ is the Heaviside function simply indicating whether a point is within the object or the background region, and $\nu = 0.6$ and $\lambda = 0.03$ are weighting parameters.

Let us take a closer look at the meaning of the three terms in the functional. The first term maximizes the a-posteriori probability of a point to belong to the assigned region. In other words: points are assigned to the region where they fit best. For a point to fit well to a region, its value must fit well to the probability density function of this region. The probability densities of the object and the background region, p_1 and p_2, are modelled as local Gaussian densities. They can be estimated given a preliminary contour and are successively updated when the contour evolves. In order to deal with textured regions, we perform the statistical modelling in the texture feature space F proposed in [6]. In case of color images it is of dimension $M = 7$. For keeping the region model manageable, the different channels are supposed to be independent, so p_1 and p_2 can be approximated by $p_i = \prod_{j=1}^{M} p_{ij}$, where p_{ij} denotes the probability density estimated in region i and channel j. Due to this statistical modelling of the regions, the contour extraction can deal with textures and is very robust under noise or other disturbances as long as one can still distinguish the regions in at least one of the image or texture channels.

The second term in (5) applies a length constraint to the contour, which effectively smoothes the contour. The amount of smoothing is determined by ν.

The last term finally takes information provided by the object model into account. The level set function Φ_0 represents the model silhouette given the current pose estimate; see Fig. 2c,d for illustration. Minimizing the distance between Φ and Φ_0 draws the contour towards the projected model, ruling out solutions that are far from its shape. Hence, pose estimation and contour extraction are coupled by this shape term: as soon as an improved pose estimate is obtained, one can compute an update of the contour and thus successively improves both the contour and the pose estimate. Each such iteration takes around 4 seconds on a 400×300 image.

Due to the sophisticated statistical region model and the integration of the object's shape, the contour can be extracted in quite general situations including background clutter, texture, and noise. However, the quality depends on a good guess of the object's pose that is involved in a) providing an initialization of Φ and b) in keeping Φ close to Φ_0.

Contour matching. Once a contour has been extracted from the image, one has to match points from this contour to 3-D points on the object surface. This is done by an iterated closest point procedure [28]. First one determines those points from the surface model that are part of the object silhouette, resulting in the 3-D object rim contour. The projection of each of these points is then matched to the closest point of the extracted contour. In this way, one obtains a 2D-3D point correspondence for each 3-D mesh point that is part of the silhouette [22, 24]. After pose estimation, a new rim contour is computed and the process is iterated until the pose converges.

These correspondences are often erroneous when the estimated pose is far from the correct pose, yet the errors tend to zero as the estimated pose gets close to the true pose. Iterating pose estimation and matching, one hopes that the estimated pose converges to the correct pose. However, the contour matching is obviously susceptible to local optima. To alleviate this problem, we use a sampling method with different (neighboring) start poses and use the resulting pose with minimum error. Depending on the number of samples, this can considerably increase the computation time, and the contour based pose tracking still stays restricted to relatively small object motions.

3.2 Optic Flow

Facing the shortcomings of contour based pose tracking, we propose the supplement of optic flow, which improves the pose tracking in two ways. Firstly, it provides additional correspondences, which makes pose estimation more robust and can resolve equivocal situations. Secondly, the object motion estimated by means of the optic flow correspondences provides a better initial guess of the pose and thus allows the tracking method to capture also large object motions.

Optic flow is defined as the 2-D vector field $\mathbf{w} := (u, v, 1)$ that matches a point in one image to the shifted point in the other image. In other words, optic flow estimation provides correspondences between the points of two images. During the past 30 years numerous techniques for optic flow estimation have emerged. Differential methods, and in particular variational methods based on the early approach of Horn and Schunck [13], are among the best performing techniques. Variational techniques combine a constancy assumption, e.g. the assumption that the gray value of a point stays constant during motion, with a smoothness assumption. Both assumptions are integrated in an energy

functional that is sought to be minimized. Thanks to the smoothness constraint, which distributes information from textured areas to close-by non-textured areas, the resulting flow field is dense, i.e., there is an optic flow estimate available for each pixel in the image.

We employ the technique from [4], which is the currently most accurate optic flow estimation method available. Let $\mathbf{x} := (x, y, t)$. Then given two images $I(x, y, t)$ and $I(x, y, t + 1)$, the technique is described by the energy minimization problem

$$E(u,v) = \underbrace{\int_\Omega \Psi\left((I(\mathbf{x}+\mathbf{w}) - I(\mathbf{x}))^2 + \gamma(\nabla I(\mathbf{x}+\mathbf{w}) - \nabla I(\mathbf{x}))^2\right)\, dxdy}_{\text{Data term}} + \alpha \underbrace{\int_\Omega \Psi(|\nabla u|^2 + |\nabla v|^2)\, dxdy}_{\text{Smoothness term}} \quad \rightarrow \min \tag{6}$$

where $\alpha = 50$ and $\gamma = 2$ are tuning parameters and $\Psi(s^2) = \sqrt{s^2 + 0.001^2}$ is a robust function which allows for outliers in both the data and the smoothness term. The data term is based on the assumption that the gray value and the gradient of a point remain constant when the point is shifted by $\mathbf{w}$. The smoothness constraint additionally requires the resulting flow field to be piecewise smooth. This optic flow estimation method has several positive properties that are important for our pose tracking task:

1. Due to non-linearized constancy assumptions, the method can deal with larger displacements than most other techniques. This ensures a good matching quality even when the object changes its pose rather rapidly.
2. It provides dense and smooth flow fields with subpixel accuracy.
3. The method is robust with respect to noise as shown in [4].
4. Thanks to the gradient constancy assumption, it is fairly robust with regard to illumination changes that appear in most real-world image sequences, e.g., due to artificial light source flickering or an automatic aperture adaptation of the camera.

Deriving 2D-3D correspondences from the optic flow. With the optic flow computed between two frames, one can establish 2D-3D point correspondences. The visible 3-D object points from the previous frame (where the pose is known) are projected to the image plane. They are then shifted according to the optic flow to their new position in the current frame. Thus, for each visible 3-D point from the last frame, one gets a correspondence to a 2-D point in the new frame.

The resulting correspondence set is used twice: firstly, it is exploited for predicting the object pose in the new frame, i.e., for getting a better pose initialization. Secondly, it is joined with the correspondence set stemming from the contour matching thus stabilizing the contour based pose estimation.

As the optic flow can also provide correspondences for points away from the rim contour of the surface, the number of correspondences is significantly larger than for the contour based matching. We therefore weight the equations (4) stemming from flow based correspondences by a factor 0.1. In this way, both correspondence sets influence the solution in an equal manner.

4 Experiments

In order to confirm the theoretical generality and robustness of the pose tracking method, it has been tested in a number of experiments using three different object models and four different image sequences.

Fig. 3 depicts an experiment where a tea box has been moved considerably between two frames. The motion is so large that the computed optic flow vectors contain errors as can be seen from the pose prediction in Fig. 3b. However, thanks to the additional contour based correspondences, the final pose result is good. Obversely, the pose estimation also fails, if only the contour based correspondences are used. This demonstrates the effective coupling of the two different ways to obtain point correspondences.

Fig. 4 depicts the coupled iteration process between contour extraction and pose estimation. As the contour evolves towards the object boundary, also the pose result improves. In return, the projected pose prohibits the contour to run away from the object in order to capture, e.g., the shadow of the tea box. Note that the setting of this experiment with a textured object, shadows, and moving background clutter rules out most alternative segmentation methods.

In the experiment depicted in Fig. 5, we tracked the pose of a quite homogeneous puncher in front of a cluttered background while the camera was moving. The camera

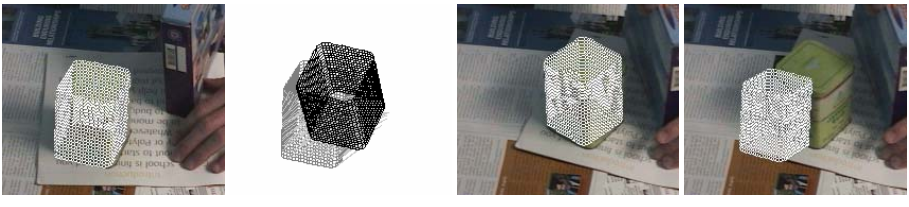

Fig. 3. The optic flow helps to capture the large motion of a tea box. **From Left to Right: (a)** Object pose at frame 1. **(b)** Object motion due to the estimated optic flow between frame 1 and frame 2. Gray: pose from frame 1. Black: pose prediction for frame 2. **(c)** Estimated pose at frame 2 using the optic flow and the evolving contour. **(d)** Estimated pose without the use of optic flow.

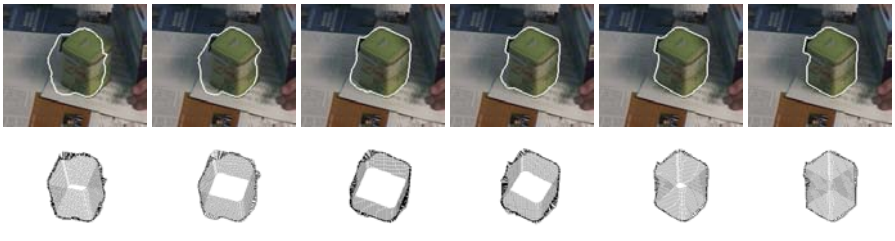

Fig. 4. Evolution of the contour and pose in Fig. 3. **From Left to Right.** Contour and pose are bad at the beginning since the optic flow estimate was erroneous, yet the contour evolves towards the object making the object pose to follow. Thereby, the shape term in the contour evolution ensures that the contour does not drift away capturing the shadow on the right.

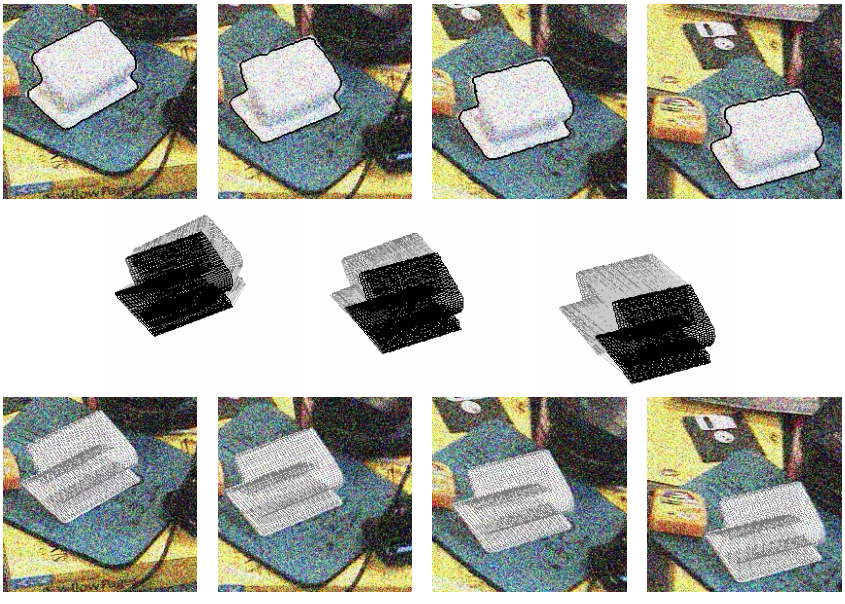

Fig. 5. Four successive frames from a sequence with the camera moving and Gaussian noise with standard deviation 60 added (242 frames, 8fps). **Top:** Extracted contour. **Center:** Object motion due to the optic flow. Gray: pose from previous frame. Black: pose prediction at current frame. **Bottom:** Estimated pose using contour and optic flow constraints.

was moved rapidly, thus the displacements between the frames are rather large and there is a motion blur in some images. Additionally, we added severe noise to the sequence. The results reveal that both the contour extraction and the optic flow estimation method can deal with these high amounts of noise. The pose prediction due to the optic flow is very good, despite the noise and the large displacements. Due to the homogeneous object surface and the noise, methods that are based on local descriptors are likely to fail in this situation.

In Fig. 6, we disturbed the puncher by adding some stickers to its surface. Since the contour extraction can deal with textured regions, also the modified puncher can be tracked accurately. One can clearly see the motion blur due to the fast camera motion. Fig. 7 shows what happens, if only flow based correspondences are used, whereas the contour based matches are neglected. Since the flow based constraints rely on the correct pose in the previous frame, errors accumulate in the course of time. This effect is avoided by the contour based correspondences.

In Fig. 8, we show the tracking result in a stereo sequence. The used tea pot model is more complex than the objects shown before. In particular, the background region is no longer connected. Here the advantage of the level set based contour representation to be able to deal with such kinds of topologies comes into play. One can see that the handle of the pot, which is quite important for a good pose estimate, is captured in three out of the four depicted images. Thanks to the integration of information from the two cameras, this works even though the hand partially occludes the handle. At

Fig. 6. Puncher disturbed by some stickers (244 frames, 8fps). **Top row:** Frames 80, 95, 100, 110, and 120. Some images show a considerable blur due to motion or the auto-focus of the camera. In others there are reflections on the puncher. **Bottom row:** Pose results at these frames.

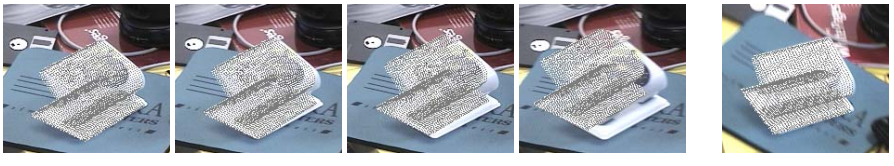

Fig. 7. Accumulation of errors when only flow based correspondences are used. **From Left to Right:** Pose results at frames 2, 5, 10, and 20. **Rightmost:** Pose result at frame 242 if contour *and* flow based correspondences are employed.

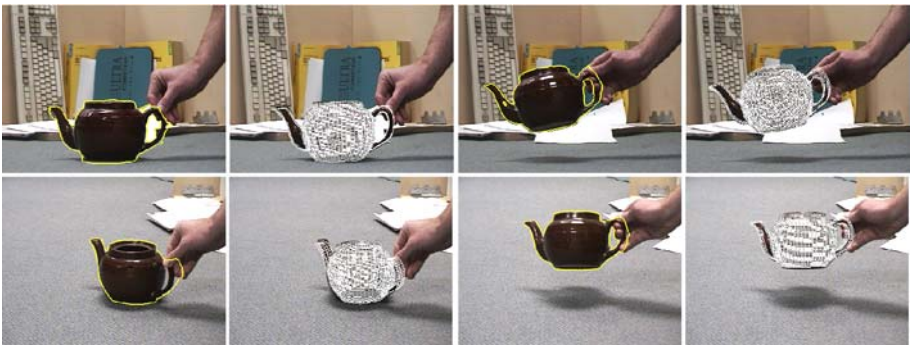

Fig. 8. Stereo sequence with partial occlusions (131 frames, 8 fps). **Top Row:** Left camera. **Bottom Row:** Right camera. **From Left to Right:** Contour at image 59. Pose at image 59. Contour at image 104. Pose at image 104.

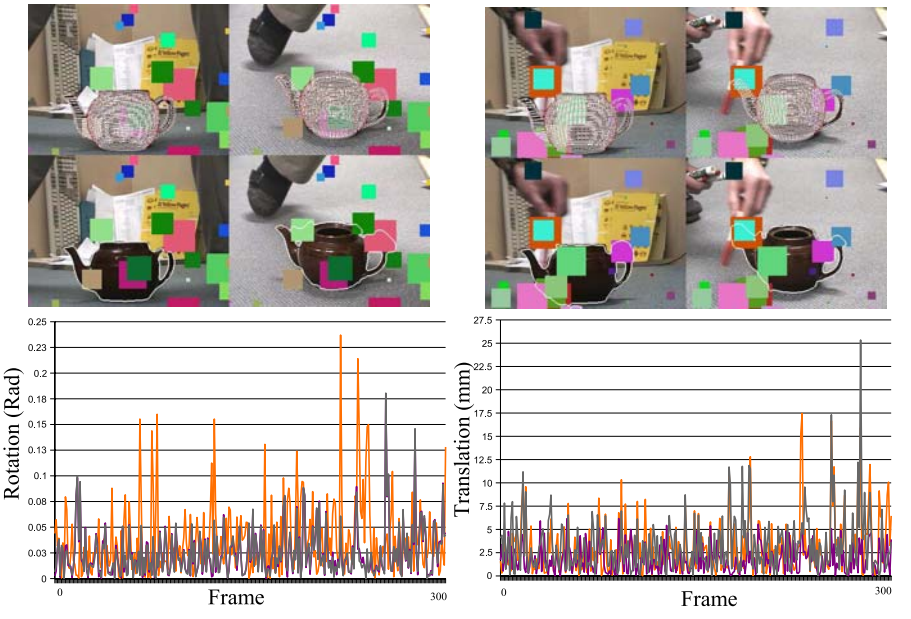

Fig. 9. Quantitative error in a static stereo scene with illumination changes and partial occlusions. The sequence has been disturbed by rectangles of random size, position, and color. **Top:** Two frames from the sequence. The right one shows the worst pose estimate according to the diagram below. **Bottom:** Rotational (left) and translational (right) errors along the three spatial axes in radians and millimeters, respectively.

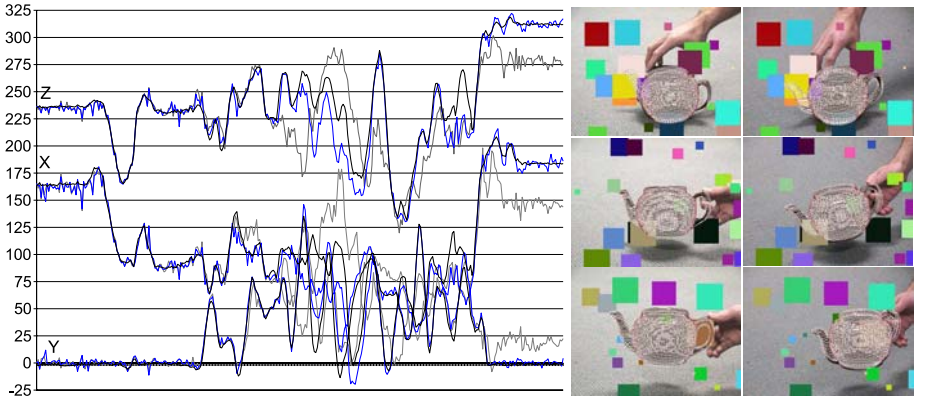

Fig. 10. Quantitative error analysis in a dynamic stereo scene disturbed by rectangles of random size, position, and color. Horizontal axis: frame number. Vertical axis: translation results (in the three spatial dimensions) **blue:** with optic flow; **gray:** without optic flow; **black:** with the undisturbed sequence. **Right:** Three stereo frames from the sequence.

this point, the optic flow providing good initializations is also very important, since the inner contour at the handle may get lost if the initialization is too far away from the correct pose.

Fig. 9 depicts a sequence where object and camera are static to allow a quantitative error measurement. The parameter settings were the same as in Fig. 8, so the object was allowed to move. The sequence has further been disturbed with rectangles of random size, position, and color which leads to occlusions of the object.

The two diagrams show the translational and angular errors along the three axes, respectively. Despite the change of the lighting conditions and partial occlusions, the error has a standard deviation of less than 7mm and 5 degrees.

Finally, Fig. 10 shows another dynamic sequence. At the beginning, the tea pot is rotated on the floor, then it is grabbed and moved around. Again the sequence has been disturbed with rectangles of random size, position, and color leading to occlusions of the object. The diagram in Fig. 10 quantifies the outcome. It shows the tracking curves for the disturbed sequence, with and without using optic flow (blue and gray, respectively) and the successful tracking of the undisturbed sequence (black) that can be regarded as some kind of ground truth. The optic flow clearly stabilizes the tracking.

The total computation time depends on the number of iterations necessary for the method to converge. For the last (and hardest) experiment we ran a setup that required approximately 2 minutes per frame on a 2.4GHz Opteron Linux machine.

5 Summary

We have suggested a pose tracking method that combines two conceptionally different matching strategies: contour matching and optic flow. Providing both qualitative and quantitative results, we have demonstrated the generality of this combination: it does not matter whether the object or the camera is moving, the method can deal with textured and homogeneous objects, as well as clutter, blurring, or noise artifacts.

In particular, we have shown that the integration of both constraints outperforms approaches that exploit only one or the other constraint. The multiresolution scheme for the optic flow estimator provides accurate contour matching even in case of larger inter-frame motion, where contour based schemes fail. The interlaced contour matching, on the other hand, prevents the accumulation of tracking errors, which is characteristic for purely optic flow based tracking systems.

References

1. A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3023 of *LNCS*, pages 54–65. Springer, May 2004.
2. H. Araújo, R. L. Carceroni, and C. M. Brown. A fully projective formulation to improve the accuracy of Lowe's pose-estimation algorithm. *Computer Vision and Image Understanding*, 70(2):227–238, May 1998.
3. C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, 2004.

4. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3024 of *LNCS*, pages 25–36. Springer, May 2004.
5. T. Brox, B. Rosenhahn, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose estimation. In W. Kropatsch, R. Sablatnig, and A. Hanbury, editors, *Pattern Recognition*, volume 3663 of *LNCS*, pages 109–116. Springer, Aug. 2005.
6. T. Brox and J. Weickert. A TV flow based local scale measure for texture discrimination. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3022 of *LNCS*, pages 578–590. Springer, May 2004.
7. T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, Feb. 2001.
8. P. David, D. DeMenthon, R. Duraiswami, and H. Samet. Simultaneous pose and correspondence determination using line features. In *Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 424–431, 2003.
9. D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99–127, July 2000.
10. P. Fua, R. Plänkers, and D. Thalmann. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding*, 81(3):285–302, Mar. 2001.
11. J. Goddard. Pose and motion estimation from vision using dual quaternion-based extended Kalman filtering. Technical report, University of Tennessee, Knoxville, 1997.
12. W. E. L. Grimson. *Object Recognition by Computer.* The MIT Press, Cambridge, MA, 1990.
13. B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
14. R. Koch. Dynamic 3D scene analysis through synthesis feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):556–568, 1993.
15. D. Lowe. Solving for the parameters of object models from image descriptions. In *Proc. ARPA Image Understanding Workshop*, pages 121–127, 1980.
16. D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
17. E. Marchand, P. Bouthemy, and F. Chaumette. A 2D-3D model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13):941–955, Nov. 2001.
18. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
19. R. Murray, Z. Li, and S. Sastry. *Mathematical Introduction to Robotic Manipulation.* CRC Press, Boca Raton, FL, 1994.
20. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
21. N. Paragios and R. Deriche. Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision. *Journal of Visual Communication and Image Representation*, 13(1/2):249–268, 2002.
22. B. Rosenhahn. Pose estimation revisited. Technical Report TR-0308, Institute of Computer Science, University of Kiel, Germany, Oct. 2003.
23. B. Rosenhahn, C. Perwass, and G. Sommer. Pose estimation of free-form contours. *International Journal of Computer Vision*, 62(3):267–289, 2005.
24. B. Rosenhahn and G. Sommer. Pose estimation of free-form objects. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3021 of *LNCS*, pages 414–427. Springer, May 2004.
25. F. Shevlin. Analysis of orientation problems using Plücker lines. In *International Conference on Pattern Recognition (ICPR)*, volume 1, pages 685–689, Brisbane, 1998.

26. L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3D camera tracking. In *3rd International Symposium on Mixed and Augmented Reality*, pages 48–57, 2004.
27. L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3D tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10): 1391–1391, 2004.
28. Z. Zang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1999.