Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

CVMR
Computer Vision & Mixed Reality Group

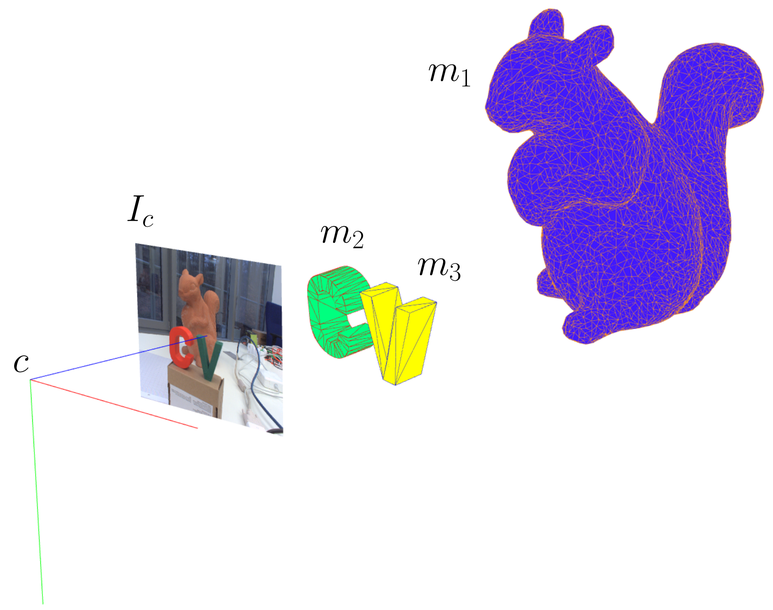
JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

Real-Time Monocular Segmentation and Pose Tracking of Multiple Objects

Henning Tjaden[1], Ulrich Schwanecke[1] and Elmar Schömer[2]

Overview

We present a real-time system capable of segmenting multiple 3D objects and tracking their pose using a single RGB camera, based on prior shape knowledge. The proposed method uses twist-coordinates for pose parametrization and a pixel-wise second-order optimization approach which lead to major improvements in terms of tracking robustness, especially in cases of fast motion and scale changes, compared to previous region-based approaches. Our C++/OpenGL implementation runs at 50–100 Hz on a commodity laptop when tracking a single object without relying on GPGPU computations.
In [1] we compare our method to the current state of the art [2] in various experiments involving challenging motion sequences and different complex objects.



Pose Parametrization

Objects $m_i, i = 1 \dots N$ are represented by dense surface models consisting of points $\mathbf{X}_m^i := (X_m^i, Y_m^i, Z_m^i)^\top \in \mathbb{R}^3$. The pre-calibrated and fixed intrinsic matrix of our camera denoted by K and the pose of an object m_i denoted by T_{cm}^i, are given by

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \text{ and } T_{cm}^i = \begin{bmatrix} R_{cm}^i & \mathbf{t}_{cm}^i \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in \mathbb{SE}(3).$$

For pose optimization we represent the rigid body motion that occurred between two consecutive frames using twists

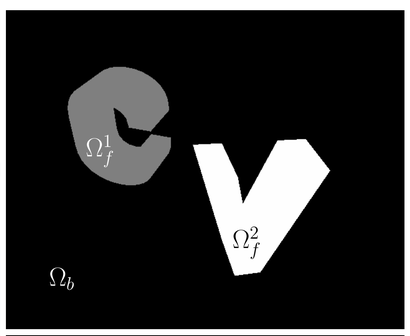
$$\theta\hat{\xi} = \theta \begin{bmatrix} \hat{\mathbf{w}} & \mathbf{v} \\ \mathbf{0}_{1\times 3} & 0 \end{bmatrix} \in \mathfrak{se}(3), \text{ with } \hat{\mathbf{w}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathfrak{so}(3),$$

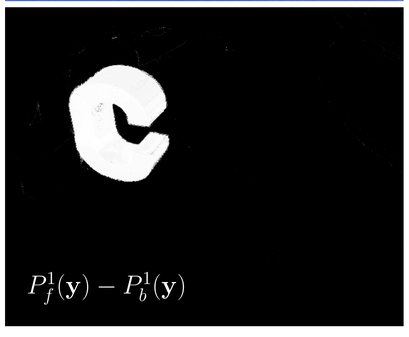
parametrized by $\theta\xi = \theta\left(\omega_1, \omega_2, \omega_3, v_1, v_2, v_2\right)^\top \in \mathbb{R}^6$, with $\mathbf{w} = (w_1, w_2, w_3)^\top$, $||\mathbf{w}||_2 = 1$, where θ is a one-parametric coupling of the rotation and translation parameters describing the motion along a screw. Each twist can be mapped to its corresponding rigid body transform via

$$exp(\theta\hat{\xi}) = \begin{bmatrix} exp(\theta\hat{\mathbf{w}}) & (\mathbb{I}_{3\times 3} - exp(\theta\hat{\mathbf{w}}))\hat{\mathbf{w}}\mathbf{v} + \mathbf{w}\mathbf{w}^\top\mathbf{v}\theta \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in \mathbb{SE}(3),$$

where $exp(\theta\hat{\mathbf{w}})$ can be computed according to Rodrigues's formula. All images are undistorted removing non-linear distortion such that the perspective projection of a surface point to an image point is given by

$$\mathbf{x}_c = \pi(K(T_{cm}\tilde{\mathbf{X}}_m)_{3\times 1}), \text{ with } \pi(\mathbf{X}) = (X/Z, Y/Z)^\top.$$





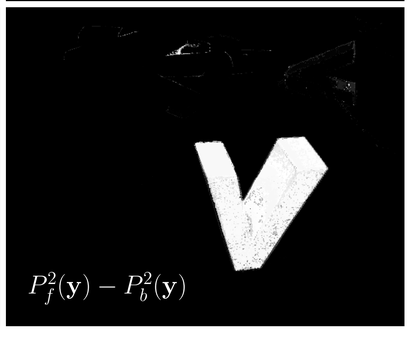
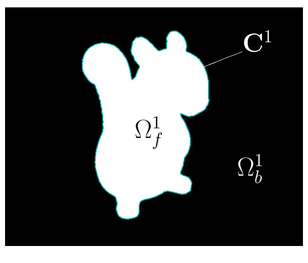


Figure 1: An illustration of pixel-wise posterior image segmentation for two different objects in the same scene with corresponding rendered silhouette masks.



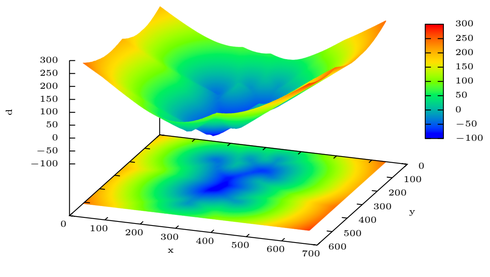


Figure 2: An illustration of the level-set pose embedding $\Phi(\mathbf{x}_c)$ applied to a rendered silhouette mask.

Pose Optimization

A camera color image is denoted by $I_c : \Omega \to \mathbb{R}^3$ where $\Omega \subset \mathbb{R}^2$ is the image domain. The color of each pixel $\mathbf{x}_c := (x_c, y_c)^\top \in \Omega$ is given by $\mathbf{y} = I_c(\mathbf{x}_c)$. Assuming pixel-wise independence we optimize the energy

$$E^i = -\sum_{\mathbf{x}_c \in \Omega} \log\Big(H_e(\Phi^i(\mathbf{x}_c))P_f^i(\mathbf{y}) + (1 - H_e(\Phi^i(\mathbf{x}_c)))P_b^i(\mathbf{y})\Big),$$

directly for the pose parameters. Here $P_f^i(\mathbf{y})$ and $P_b^i(\mathbf{y})$ represent the foreground and background region membership probability per object of each pixel's color (see Figure 1) and Φ^i is the level-set embedding of each object's contour defined by its pose (see Figure 2). Thereby, the gradient of the energy is given by

$$\frac{\partial E^i(\theta\xi)}{\partial\theta\xi} = -\sum_{\mathbf{x}_c \in \Omega} \frac{P_f(\mathbf{y}) - P_b(\mathbf{y})}{H_e(\Phi^i(\mathbf{x}_c, \theta\xi))(P_f(\mathbf{y}) - P_b(\mathbf{y})) + P_b(\mathbf{y})} \cdot \delta_e \cdot \frac{\partial\Phi^i(\mathbf{x}_c, \theta\xi)}{\partial\theta\xi}$$

where δ_e is the smoothed Dirac delta function corresponding to H_e. Assuming small motion we get

$$\frac{\partial\Phi^i(\mathbf{x}_c, \theta\xi_0)}{\partial\theta\xi} = \left[\frac{\partial\Phi^i}{\partial x_c}, \frac{\partial\Phi^i}{\partial y_c}\right] \begin{bmatrix} \frac{f_x}{Z_c^i} & 0 & -\frac{X_c^i f_x}{(Z_c^i)^2} \\ 0 & \frac{f_y}{Z_c^i} & -\frac{Y_c^i f_y}{(Z_c^i)^2} \end{bmatrix} \begin{bmatrix} 0 & Z_c^i & -Y_c^i & 1 & 0 & 0 \\ -Z_c^i & 0 & X_c^i & 0 & 1 & 0 \\ Y_c^i & -X_c^i & 0 & 0 & 0 & 1 \end{bmatrix},$$

with $\mathbf{X}_c^i = (X_c^i, Y_c^i, Z_c^i)^\top = (T_{cm}^i\tilde{\mathbf{X}}_m^i)_{3\times 1}$. This results in the 1×6 per pixel Jacobian $J^i(\mathbf{x}_c, \theta\xi_0) = \partial E^i(\mathbf{x}_c, \theta\xi_0)/\partial\theta\xi$, evaluated at $\theta\xi_0 = \mathbf{0}^\top$. At each iteration the twist parameter step is calculated as

$$\Delta\theta\xi^i = -\Big(\sum_{\mathbf{x}_c \in \Omega} J^i(\mathbf{x}_c, \theta\xi_0)^\top J^i(\mathbf{x}_c, \theta\xi_0)\Big)^{-1} \sum_{\mathbf{x}_c \in \Omega} J^i(\mathbf{x}_c, \theta\xi_0)^\top,$$

using Cholesky decomposition. The resulting step is mapped to its corresponding rigid body transform and applied to the initial transform estimate as

$$T_{cm}^i \leftarrow exp(\Delta\theta\xi^i)T_{cm}^i.$$

To improve robustness and runtime, we compute this pose optimization in a coarse to fine manner.

Occlusion Handling

When tracking multiple objects simultaneously, mutual occlusions are very likely to emerge, which must be handled appropriately. In order to minimize rendering and memory transfer we render the entire scene once per iteration, download a common silhouette mask I_s and the according depth-buffer I_d. For each model m_i we then compute Φ^i directly from I_s. Thereby, the respective contours C^i can contain segments resulting from occlusions that are considered in the respective signed distance transform. To handle this, for each object all pixels with a distance value that was influenced by occlusion have to be discarded for pose optimization (see Figure 3). The performance of our strategy is demonstrated in an example experiment (see Figure 4).

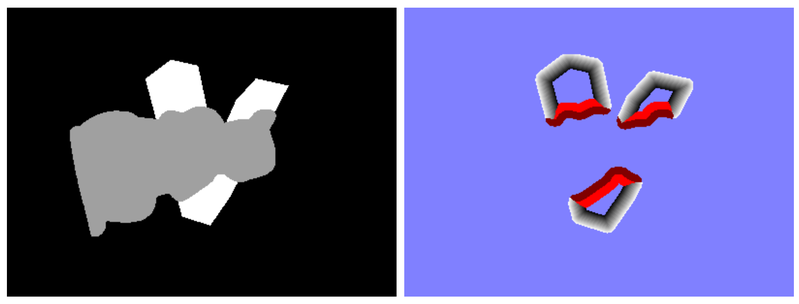


Figure 3: Multiple object tracking with occlusion. Left: Common silhouette mask I_s of corresponding surface models m_1 and m_2. Right: Φ^2 within ± 8 pixels around the occluded contour C^2 (grey values) and pixels influenced by occlusion (bright red inside, dark red outside of Ω_f^2).

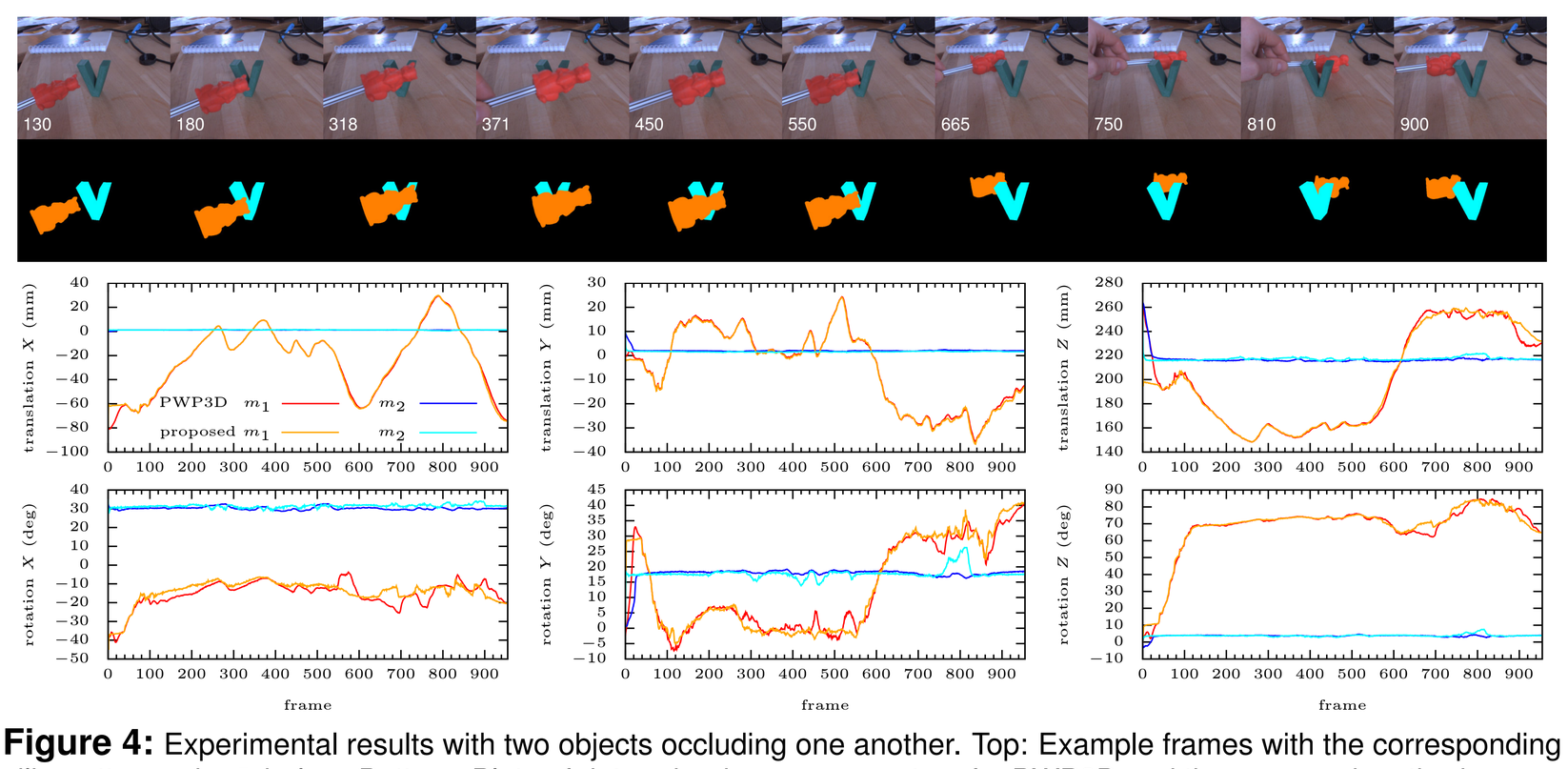


Figure 4: Experimental results with two objects occluding one another. Top: Example frames with the corresponding silhouette masks I_s below. Bottom: Plots of determined pose parameters for PWP3D and the proposed method.

References

[1] Henning Tjaden, Ulrich Schwanecke, and Elmar Schömer.
Real-time monocular segmentation and pose tracking of multiple objects.
In *14th European Conference on Computer Vision (ECCV)*, October 2016.

[2] Victor A. Prisacariu and Ian D. Reid.
Pwp3d: Real-time segmentation and tracking of 3d objects.
International Journal of Computer Vision, 98(3):335–354, July 2012.

[1]Computer Science Department, RheinMain University of Applied Sciences

[2]Institute of Computer Science, Johannes Gutenberg University Mainz