Real-Time 3D Model Tracking in Color and Depth on a Single CPU Core

Wadim Kehl * Federico Tombari † Slobodan Ilic ♦ Nassir Navab †

* Toyota Research Institute, Los Altos, USA † Technical University of Munich, Germany ♦ Siemens Research, Munich, Germany

1. Introduction

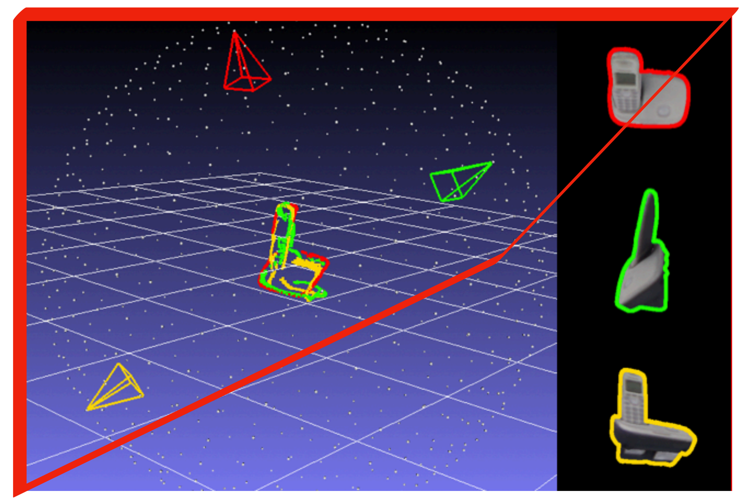
We present a novel formulation for 3D model tracking in color and depth data via a joint error over both the contour and an ICP energy

$$E_{Joint} = E_C + \lambda E_{ICP}$$

We can deal with typical problems such as multiple instance tracking, occlusion and scale changes

2. Viewpoint approximation

To speed up computation, we introduce approximations that enable us to track many objects in real-time on a single core.

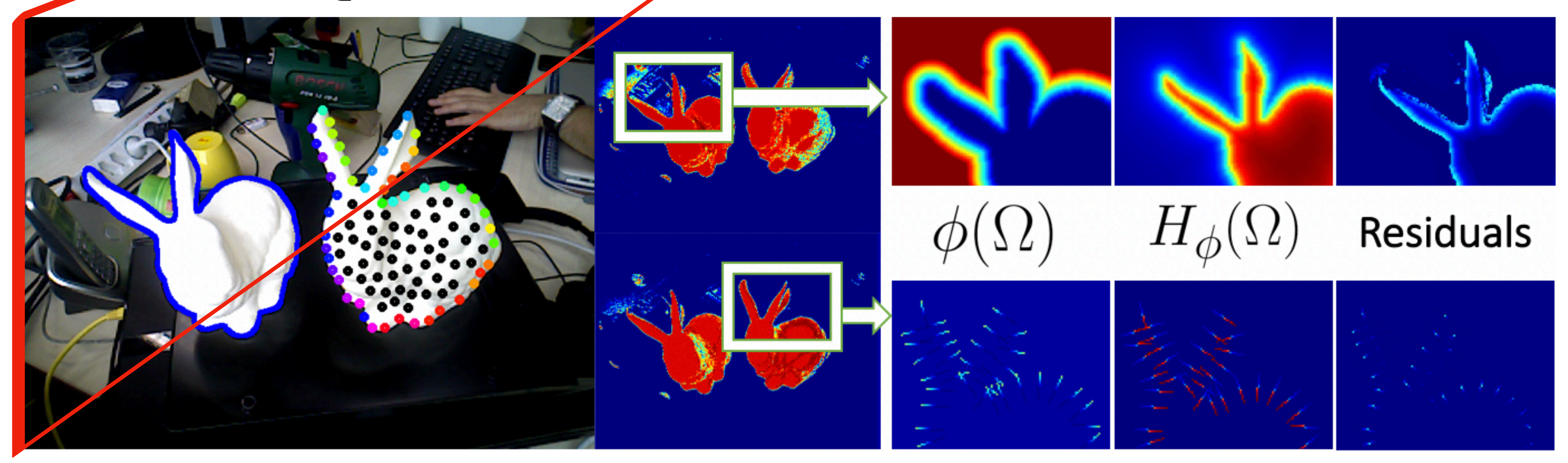


By pre-rendering the object from densely sampled viewpoints, we extract scale-invariant information such as 3D contour points and interior surface points. This allows us to avoid any GPU involvement during tracking.

3a. Contour energy

Following PWP3D [1], we define the pixel-wise posterior via foreground/ background probabilities and the Heaviside function of an implicit contour. We then sum over the negative logs to finally retrieve:

$$E_C := -\sum_{x \in \Omega} \log \Big(H_\phi(x) P_f(I(x)) + (1 - H_\phi(x)) P_b(I(x)) \Big)$$



Tracking two Stanford bunnies side by side. While the left bunny is tracked densely [1], our approach evaluates the energy using the sparse 3D contour points extracted in the offline stage. We can thus both circumvent a rendering step and a distance transform.

[1] V. A. Prisacariu and I. D. Reid. PWP3D: Real-Time Segmentation and Tracking of 3D Objects. IJCV, 2012.

3b. ICP energy

We define a 'plane-to-point' ICP energy, where the normals are coming from the source and must be rotated as well. Since we pre-computed our viewpoints, we can use the 3D normals of the sparse interior surface points without additional costs.

$$E_{ICP} := \underset{\Xi}{\operatorname{argmin}} \sum_i \Big((\Xi(\bar{s}_i) - d_i) \cdot \Xi_{SO}(\bar{n}_i) \Big)$$

We transform the local 3D information into the scene

$$\bar{s}_i = R \cdot s_i + t \qquad \bar{n}_i = R \cdot n_i,$$

and find the projective correspondence $d_i := \Pi_D^{-1}(\pi(\bar{s}_i))$

4. Optimization

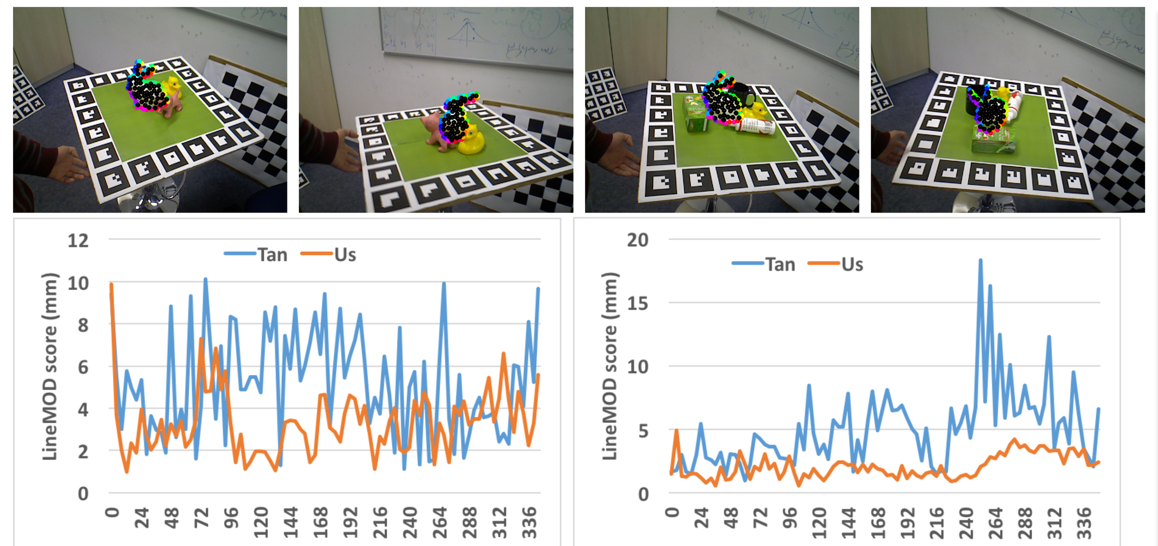
By deriving in respect to the object pose, we retrieve both the separate terms for the contour over pixels and the ICP energy over 3D correspondences:

$$\frac{\partial E_C}{\partial \xi} = \frac{(P_f - P_b)}{H_\phi (P_f - P_b) + P_b} \frac{\partial H_\phi}{\partial \phi} \frac{\partial \phi}{\partial x} \frac{\partial \pi(X)}{\partial X} \frac{\partial \Xi(X)}{\partial \xi}$$

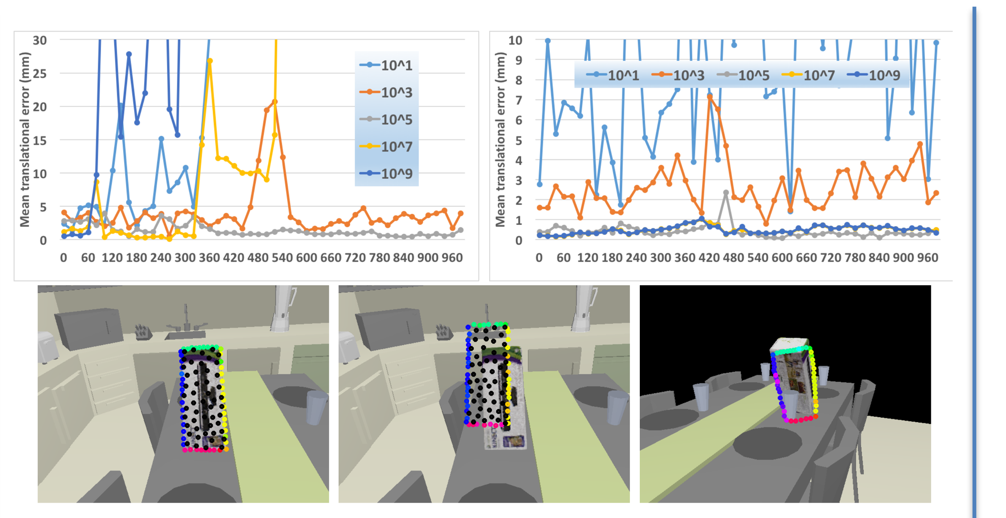
$$J_i := -\left[\bar{n}_i^\intercal \quad \Big((\bar{s}_i \times \bar{n}_i) + \bar{n}_i \times (\bar{s}_i - d_i) \Big)^\intercal \right] \qquad r_i := (\bar{s}_i - d_i) \cdot \bar{n}_i$$

The final twist is then computed as a weighted update between those two energies of pixel-wise and correspondence-wise errors

$$\nabla \xi = \left(\boxed{\sum_x J_x^\intercal J_x} + \boxed{\sum_i \lambda J_i^\intercal J_i} \right)^{-1} \left(\boxed{\sum_x J_x} + \boxed{\sum_i \sqrt{\lambda} J_i \cdot r_i} \right)$$



Evaluation of tracking error on custom sequences of [3]. We constantly demonstrate better performance.



Evaluation of weighting parameter λ on two sequences of [2]. The object properties decide which configuration is best for tracking.

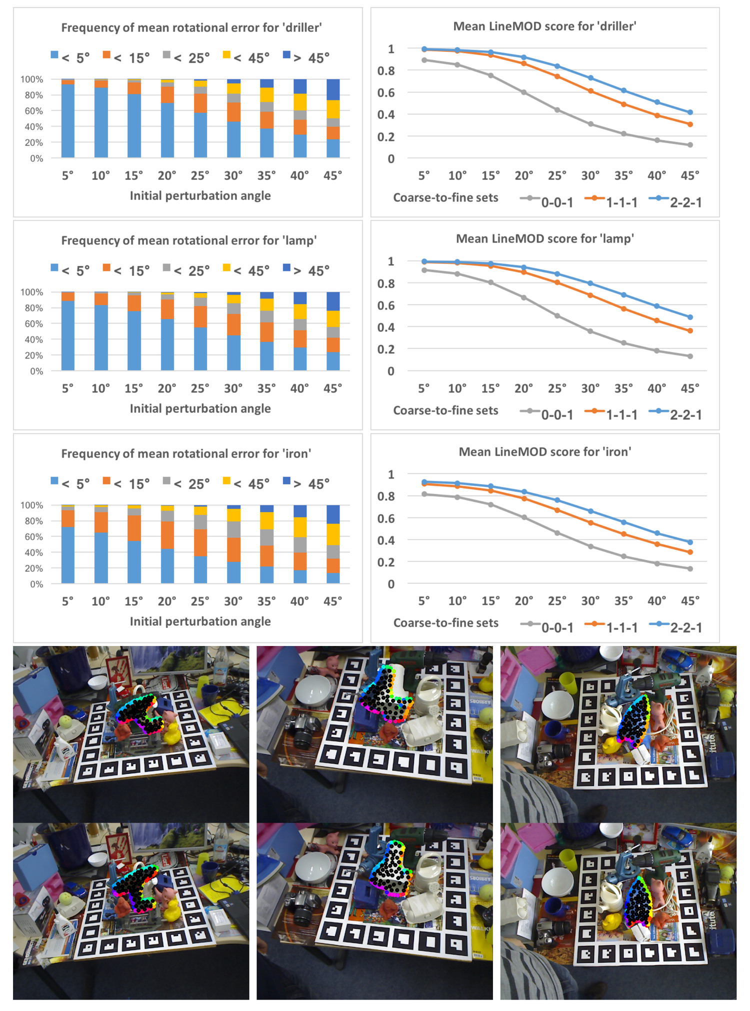
[2] C. Choi and H. Christensen. RGB-D Object Tracking: A Particle Filter Approach on GPU. In IROS, 2013

[3] D. J. Tan, F. Tombari, S. Ilic, and N. Navab. A Versatile Learning-based 3D Temporal Tracker: Scalable, Robust, Online. In ICCV, 2015.

5. Results

		PCL	C&C	Krull	Tan	A	B
(a) Kinect Box	t_x	43.99	1.84	0.8	1.54	1.2	**0.76**
	t_y	42.51	2.23	1.67	1.90	1.16	**1.09**
	t_z	55.89	1.36	0.79	0.34	**0.30**	0.38
	α	7.62	6.41	1.11	0.42	**0.14**	0.17
	β	1.87	0.76	0.55	0.22	0.23	**0.18**
	γ	8.31	6.32	1.04	0.68	0.22	**0.20**
	ms	4539	166	143	**1.5**	2.70	8.10
(b) Milk	t_x	13.38	0.93	**0.51**	1.23	0.91	0.64
	t_y	31.45	1.94	1.27	0.74	0.71	**0.59**
	t_z	26.09	1.09	0.62	**0.24**	0.26	**0.24**
	α	59.37	3.83	2.19	0.50	0.44	**0.41**
	β	19.58	1.41	1.44	**0.28**	0.31	0.29
	γ	75.03	3.26	1.90	0.46	0.43	**0.42**
	ms	2205	134	135	**1.5**	2.72	8.54
(c) Orange Juice	t_x	2.53	0.96	0.52	1.10	0.59	**0.50**
	t_y	2.20	1.44	0.74	0.94	**0.64**	0.69
	t_z	1.91	1.17	0.63	0.18	0.18	**0.17**
	α	85.81	1.32	1.28	0.35	**0.12**	**0.12**
	β	42.12	0.75	1.08	0.24	0.22	**0.20**
	γ	46.37	1.39	1.20	0.37	**0.18**	0.19
	ms	1637	117	129	**1.5**	2.79	8.79
(d) Tide	t_x	1.46	0.83	0.69	0.73	0.36	**0.34**
	t_y	2.25	1.37	0.81	0.56	0.51	**0.49**
	t_z	0.92	1.20	0.81	0.24	**0.18**	**0.18**
	α	5.15	1.78	2.10	0.31	0.20	**0.15**
	β	2.13	1.09	1.38	**0.25**	0.43	0.39
	γ	2.98	1.13	1.27	**0.34**	0.39	0.37
	ms	2762	111	116	**1.5**	2.71	9.42
Mean	Tra	18.72	1.36	0.82	0.81	0.58	**0.51**
	Rot	29.70	2.45	1.38	0.37	0.28	**0.26**
	ms	2786	132	131	**1.5**	2.73	8.71

Comparison to related work on [2].



Convergence experiments to highlight the recovery abilities of our method for imperfect initializations

Tracking error for changing number of sample points, i.e. the degree of approximation. Even with a only a few points, tracking is possible!

Acknowledgements

The authors would like to thank Toyota Motor Corporation for funding and support as well as Henning Tjaden for fruitful discussions.