Iterative Corresponding Geometry: Fusing Region and Depth for Highly Efficient 3D Tracking of Textureless Objects – Supplementary

Manuel Stoiber[1,2] Martin Sundermeyer[1,2] Rudolph Triebel[1,2]
[1] German Aerospace Center (DLR) [2] Technical University of Munich (TUM)
{firstname.lastname}@dlr.de

1. Introduction

In the following, we first state timings for the individual steps of our algorithm. After this, the full results on the Choi dataset [1] are presented, for which a concise version was shown in the paper. Subsequently, the developed region modality is evaluated on the *RBOT* dataset [16], demonstrating improved tracking success. Also, we compare to state-of-the-art 6DoF pose estimation algorithms on the *YCB-Video* dataset [18] and discuss the role of 3D object tracking. Finally, using predictions from modern pose estimation algorithms, we demonstrate that *ICG* is well-suited for highly efficient pose refinement.

2. Timings

In our work, an average framerate of $788.4\,\mathrm{Hz}$ was given for the evaluation on the *YCB-Video* dataset [18]. This corresponds to a total duration of $1.27\,\mathrm{ms}$ per frame. Of this time, the algorithm needs $0.52\,\mathrm{ms}$ for the computation of correspondence lines, $0.58\,\mathrm{ms}$ for correspondence points, $0.09\,\mathrm{ms}$ for the calculation of gradient vectors and Hessian matrices, $0.05\,\mathrm{ms}$ for the update of color histograms, and the remaining $0.03\,\mathrm{ms}$ for other operations such as the optimization and pose update. The timings demonstrate that the region- and depth-modality are well balanced, requiring similar amounts of computation.

3. Choi Dataset

In the paper, only the averages over rotational and translational RMS errors were presented for the Choi dataset [1]. For the sake of completeness, we also want to provide the full results with respect to the errors in the x, y, and z directions and in the roll, pitch, and yaw angles. The results for each of the four evaluated objects as well as the mean values are shown in Tab. 1.

Table 1. RMS errors for translation and rotation parameters on the Choi dataset [1]. Results are from the respective papers.

Approach		Choi [1]	Krull [7]	Tan [15]	Kehl [6]	ICG (Ours)
Kinect Box	X	1.84	0.83	0.15	0.76	**0.05**
	Y	2.23	1.67	0.19	1.09	**0.11**
	Z	1.36	0.79	0.09	0.38	**0.03**
	Roll	6.41	1.11	0.09	0.17	**0.02**
	Pitch	0.76	0.55	0.06	0.18	**0.02**
	Yaw	6.32	1.04	0.04	0.20	**0.02**
Milk	X	0.93	0.51	0.09	0.64	**0.02**
	Y	1.94	1.27	0.11	0.59	**0.05**
	Z	1.09	0.62	0.08	0.24	**0.02**
	Roll	3.83	2.19	0.07	0.41	**0.06**
	Pitch	1.41	1.44	0.09	0.29	**0.04**
	Yaw	3.26	1.90	**0.06**	0.42	**0.06**
Orange Juice	X	0.96	0.52	0.11	0.50	**0.04**
	Y	1.44	0.74	0.09	0.69	**0.03**
	Z	1.17	0.63	0.09	0.17	**0.02**
	Roll	1.32	1.28	0.08	0.12	**0.05**
	Pitch	0.75	1.08	0.08	0.20	**0.03**
	Yaw	1.39	1.20	0.08	0.19	**0.06**
Tide	X	0.83	0.69	0.08	0.34	**0.02**
	Y	1.37	0.81	0.09	0.49	**0.03**
	Z	1.20	0.81	0.07	0.18	**0.01**
	Roll	1.78	2.10	0.05	0.15	**0.03**
	Pitch	1.09	1.38	0.12	0.39	**0.04**
	Yaw	1.13	1.27	0.05	0.37	**0.03**
Mean Translation		1.36	0.82	0.10	0.51	**0.04**
Mean Rotation		2.45	1.38	0.07	0.26	**0.04**

4. RBOT Dataset

In our work, we modified the region-based approach of *SRT3D* [11, 12] to be independent of the scale space and to incorporate a user-defined uncertainty. In the following, we want to show that this is not only convenient for the combination with the depth modality but that the modifications also improve tracking results. We thereby use the *RBOT* dataset [16] to compare our approach to the state of the art in region-based tracking as well as to additional methods that include edge information.

All experiments in the evaluation are performed as defined by [16]. The required translational and rotational er-

Table 2. Tracking success rate for all objects and scenarios on the *RBOT* dataset [16]. Methods that incorporate information from edges in addition to region are indicated by a *. Results are from the respective publications.

Approach	Ape	Soda	Vise	Soup	Camera	Can	Cat	Clown	Cube	Driller	Duck	Egg Box	Glue	Iron	Candy	Lamp	Phone	Squirrel	**Avg.**
Regular																			
Tjaden [16]	85.0	39.0	98.9	82.4	79.7	87.6	95.9	93.3	78.1	93.0	86.8	74.6	38.9	81.0	46.8	97.5	80.7	99.4	79.9
Zhong [19]	88.8	41.3	94.0	85.9	86.9	89.0	98.5	93.7	83.1	87.3	86.2	78.5	58.6	86.3	57.9	91.7	85.0	96.2	82.7
Huang [5]*	91.9	44.8	**99.7**	89.1	89.3	90.6	97.4	95.9	83.9	97.6	91.8	84.4	59.0	92.5	74.3	97.4	86.4	99.7	86.9
Liu [10]*	93.7	39.3	98.4	91.6	84.6	89.2	97.9	95.9	86.3	95.1	93.4	77.7	61.5	87.8	65.0	95.2	85.7	99.8	85.5
Li [9]*	92.8	42.6	96.8	87.5	90.7	86.2	99.0	96.9	86.8	94.6	90.4	87.0	57.6	88.7	59.9	96.5	90.6	99.5	85.8
Sun [13]*	93.0	55.2	99.3	85.4	96.1	93.9	98.0	95.6	79.5	**98.2**	89.7	89.1	66.5	91.3	60.6	**98.6**	95.6	99.6	88.1
SRT3D [12]	**98.8**	65.1	99.6	**96.0**	**98.0**	96.5	**100.0**	98.4	94.1	96.9	**98.0**	95.3	79.3	**96.0**	90.3	97.4	96.2	99.8	94.2
ICG (Ours)	98.1	**66.4**	99.6	**96.0**	97.4	**96.9**	**100.0**	**98.5**	**94.8**	97.6	**98.0**	**95.5**	**80.8**	95.9	**91.0**	97.1	**96.6**	**99.9**	**94.4**
Dynamic Light																			
Tjaden [16]	84.9	42.0	99.0	81.3	84.3	88.9	95.6	92.5	77.5	94.6	86.4	77.3	52.9	77.9	47.9	96.9	81.7	99.3	81.2
Zhong [19]	89.7	40.2	92.7	86.5	86.6	89.2	98.3	93.9	81.8	88.4	83.9	76.8	55.3	79.3	54.7	88.7	81.0	95.8	81.3
Huang [5]*	91.8	42.3	98.9	89.9	91.3	87.8	97.6	94.5	84.5	98.1	91.9	86.7	66.2	90.9	73.2	97.1	89.2	99.6	87.3
Liu [10]*	93.5	38.2	98.4	88.8	87.0	88.5	98.1	94.4	85.1	95.1	92.7	76.1	58.1	79.6	62.1	93.2	84.7	99.6	84.1
Li [9]*	93.5	43.1	96.6	88.5	92.8	86.0	99.6	95.5	85.7	96.8	91.1	90.2	68.4	86.8	59.7	96.1	91.5	99.2	86.7
Sun [13]*	93.8	55.9	**99.6**	85.6	**97.7**	93.7	97.7	96.5	78.3	**98.6**	91.0	91.6	72.1	90.7	63.0	**98.9**	94.4	**100.0**	88.8
SRT3D [12]	98.2	65.2	99.2	95.6	97.5	**98.1**	**100.0**	98.5	94.2	97.5	**97.9**	96.9	**86.1**	95.2	89.3	97.0	95.9	99.9	94.6
ICG (Ours)	**98.4**	**67.0**	99.5	**95.7**	97.6	97.5	99.8	**98.6**	**94.9**	97.5	97.4	**97.1**	85.5	**96.0**	**91.5**	97.7	**96.2**	99.9	**94.9**
Noise																			
Tjaden [16]	77.5	44.5	91.5	82.9	51.7	38.4	95.1	69.2	24.4	64.3	88.5	11.2	2.9	46.7	32.7	57.3	44.1	96.6	56.6
Zhong [19]	79.3	35.2	82.6	86.2	65.1	56.9	96.9	67.0	37.5	75.2	85.4	35.2	18.9	63.7	35.4	64.6	66.3	93.2	63.6
Huang [5]*	89.0	45.0	89.5	90.2	68.9	38.3	95.9	72.8	20.1	85.5	92.2	26.8	15.8	66.2	52.2	58.3	65.1	98.4	65.0
Liu [10]*	84.7	33.0	88.8	89.5	56.4	50.1	94.1	66.5	32.3	79.6	94.2	29.6	19.9	63.4	40.3	61.6	62.4	96.9	63.5
Li [9]*	89.1	44.0	91.6	89.4	75.2	62.3	98.6	77.3	41.2	81.5	91.6	54.5	31.8	65.0	46.0	78.5	69.6	97.6	71.4
Sun [13]*	92.5	56.2	**98.0**	85.1	**91.7**	**79.0**	97.7	86.2	40.1	**96.6**	90.8	**70.2**	**50.9**	**84.3**	49.9	**91.2**	**89.4**	99.4	80.5
SRT3D [12]	96.9	61.9	95.4	95.7	84.5	73.9	**99.9**	90.3	**62.2**	87.8	**97.6**	62.2	43.4	**84.3**	78.2	73.3	83.1	**99.7**	81.7
ICG (Ours)	**98.0**	**64.3**	95.4	**95.8**	84.8	74.8	**99.9**	**90.5**	61.9	88.5	97.4	63.4	45.3	84.2	**81.2**	74.0	84.8	99.4	**82.4**
Unmodeled Occlusion																			
Tjaden [16]	80.0	42.7	91.8	73.5	76.1	81.7	89.8	82.6	68.7	86.7	80.5	67.0	46.6	64.0	43.6	88.8	68.6	86.2	73.3
Zhong [19]	83.9	38.1	92.4	81.5	81.3	85.5	97.5	88.9	76.1	87.5	81.7	72.7	52.5	77.2	53.9	88.5	79.3	92.5	78.4
Huang [5]*	86.2	46.3	97.8	87.5	86.5	86.3	95.7	90.7	78.8	96.5	86.0	80.6	59.9	86.8	69.6	93.3	81.8	95.8	83.6
Liu [10]*	87.1	36.7	91.7	78.8	79.2	82.5	92.8	86.1	78.0	90.2	83.4	72.0	52.3	72.8	55.9	86.9	77.8	93.0	77.6
Li [9]*	89.3	43.3	92.2	83.1	84.1	79.0	94.5	88.6	76.2	90.4	87.0	80.7	61.6	75.3	53.1	91.1	81.9	93.4	80.3
Sun [13]*	91.3	56.7	97.8	82.0	92.8	89.9	96.6	92.2	71.8	**97.0**	85.0	84.6	66.9	87.7	56.1	95.1	89.8	98.2	85.1
SRT3D [12]	96.5	**66.8**	99.0	95.8	**95.0**	95.9	**100.0**	97.6	92.2	96.6	95.0	94.4	79.0	94.7	**89.8**	95.7	93.6	**99.6**	93.2
ICG (Ours)	**97.3**	66.3	**99.3**	**96.0**	**95.0**	**96.5**	**100.0**	**97.7**	**92.9**	96.4	**96.1**	**96.5**	**82.1**	**96.1**	89.7	**95.8**	**94.2**	99.2	**93.7**
Modeled Occlusion																			
Tjaden [16]	82.0	42.0	95.7	81.1	78.7	83.4	92.8	87.9	74.3	91.7	84.8	71.0	49.1	73.0	46.3	90.9	76.2	96.9	77.7
Huang [5]*	87.8	45.5	98.1	87.2	89.0	89.8	95.1	91.4	77.4	**97.1**	87.7	83.0	62.5	88.6	69.7	94.1	86.0	98.9	84.9
SRT3D [12]	**97.9**	68.3	99.2	95.4	96.8	96.4	99.6	98.6	93.0	96.4	96.6	96.2	82.9	95.1	91.0	96.0	94.5	99.6	94.1
ICG (Ours)	**97.9**	**69.1**	**99.5**	**97.2**	**97.1**	**96.9**	**99.9**	**98.9**	**93.2**	97.0	**97.8**	**97.2**	**84.3**	**96.0**	**92.6**	**97.4**	**95.3**	**99.8**	**94.8**

rors are calculated as follows

$$e_{\mathrm{t}} = \left\| {}_{\mathrm{M}}\boldsymbol{t}_{\mathrm{M_{gt}}} \right\|_2, \tag{1}$$

$$e_{\mathrm{r}} = \cos^{-1}\left(\frac{\mathrm{trace}({}_{\mathrm{M}}\boldsymbol{R}_{\mathrm{M_{gt}}}) - 1}{2} \right). \tag{2}$$

Based on those errors, the tracking success is calculated as the percentage of estimated poses with $e_{\mathrm{t}} < 5\,\mathrm{cm}$ and $e_{\mathrm{r}} < 5^\circ$. In cases of unsuccessful tracking, the algorithm is re-initialized with the ground-truth pose. For *ICG*, we use the same parameter values as in [12] and define a decreasing standard deviation of $\sigma_{\mathrm{r}} = \{15, 5, 3.5, 1.5\}$.

Results of the evaluation are shown in Tab. 2. The reported scores show that both *SRT3D* and *ICG* achieve significantly higher results than the remaining algorithms. However, on average, *ICG* performs about half a percentage point better than *SRT3D*. This demonstrates that setting a defined standard deviation σ_{r} instead of using an implicit variance of $\sigma^2 = {s_{\mathrm{h}}}^2/\bar{n}^2$ helps to further improve results.

5. 6DoF Pose Estimation

Given the strong results of modern 6DoF pose estimation methods [2, 8], the question arises whether 3D object tracking is even necessary. In order to answer this, we compare *ICG* with state-of-the-art pose estimation methods on the *YCB-Video* dataset [18]. The *ADD(S)* metric is thereby adopted to ensure compatibility with reported results from *PVN3D* [3] and *FFB6D* [2]. It is a combined metric that uses the *ADD-S* score for symmetric objects and the *ADD* error in all other cases.

Results of the evaluation are shown in Tab. 3. The comparison demonstrates that *ICG* is able to outperform most methods by a considerable margin for the *ADD-S* score, per-

Table 3. Comparison against state-of-the-art 6DoF pose estimation methods on the *YCB-Video* dataset [18] with *ADD(S)* and *ADD-S* area under curve scores in percent. Results for *Augmented Autoencoders*[2] [14], *CosyPose*[3] [8], and *ICG* were computed by us. All other values are from the respective publications. Note that while *CosyPose* was trained on real data, good results can also be obtained using synthetic data alone [4]. Symmetric objects for which the *ADD(S)* metric reports the *ADD-S* instead of the *ADD* error are indicated by a ⋆.

Approach	PoseCNN [18]		Augmented Autoencoders[2] [14]		DenseFusion [17]		CosyPose[3] [8]		PVN3D [3]		FFB6D [2]		ICG (Ours)	
(Training) Data	Real RGB		3D Model		Real RGB-D		Real RGB		Real RGB-D		Real RGB-D		3D Model	
Objects	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S	ADD(S)	ADD-S
002_master_chef_can	50.9	84.0	27.1	50.6	-	**96.4**	37.3	90.6	80.5	96.0	**80.6**	96.3	66.4	89.7
003_cracker_box	51.7	76.9	32.2	64.5	-	95.5	76.8	94.9	**94.8**	96.1	94.6	**96.3**	82.4	92.1
004_sugar_box	68.6	84.3	73.6	88.6	-	97.5	95.2	97.6	96.3	97.4	**96.6**	97.6	96.1	**98.4**
005_tomato_soup_can	66.0	80.9	72.3	84.4	-	94.6	**90.5**	94.6	88.5	96.2	89.6	95.6	73.2	**97.3**
006_mustard_bottle	79.9	90.2	77.5	90.9	-	97.2	92.7	96.5	96.2	97.5	**97.0**	97.8	96.2	**98.4**
007_tuna_fish_can	70.4	87.9	71.2	92.2	-	96.6	**93.9**	**97.5**	89.3	96.0	88.9	96.8	73.2	95.8
008_pudding_box	62.9	79.0	47.9	67.7	-	96.5	93.5	96.2	**95.7**	**97.1**	94.6	**97.1**	73.8	88.9
009_gelatin_box	75.2	87.1	74.8	82.9	-	98.1	94.1	96.1	96.1	97.7	96.9	98.1	**97.2**	**98.8**
010_potted_meat_can	59.6	78.5	53.6	63.3	-	91.3	75.9	84.0	88.6	93.3	88.1	94.7	**93.3**	**97.3**
011_banana	72.3	85.9	13.1	51.6	-	96.6	90.0	95.6	93.7	96.6	94.9	97.2	**95.6**	**98.4**
019_pitcher_base	52.5	76.8	77.6	91.7	-	97.1	94.0	97.3	96.5	97.4	96.9	97.6	**97.0**	**98.8**
021_bleach_cleanser	50.5	71.9	42.0	62.6	-	95.8	82.1	92.7	93.2	96.0	**94.8**	96.8	92.6	**97.5**
024_bowl⋆	69.7	69.7	79.1	79.1	-	88.2	87.8	87.8	90.2	90.2	96.3	96.3	**98.4**	**98.4**
025_mug	57.7	78.0	58.0	80.9	-	97.1	87.8	94.9	95.4	97.6	94.2	97.3	**95.6**	**98.5**
035_power_drill	55.1	72.8	61.2	77.9	-	96.0	89.7	95.1	95.1	96.7	95.9	97.2	**96.7**	**98.5**
036_wood_block⋆	65.8	65.8	55.2	55.2	-	89.7	80.5	80.5	90.4	90.4	92.6	92.6	**97.2**	**97.2**
037_scissors	35.8	56.2	0.8	7.0	-	95.2	67.6	81.5	92.7	96.7	**95.7**	**97.7**	93.5	97.3
040_large_marker	58.0	71.4	55.6	67.6	-	97.5	84.3	93.1	**91.8**	96.7	89.1	96.6	88.5	**97.8**
051_large_clamp⋆	49.9	49.9	72.2	72.2	-	72.9	91.3	91.3	93.6	93.6	96.8	96.8	**96.9**	**96.9**
052_extra_large_clamp⋆	47.0	47.0	59.5	59.5	-	69.8	75.7	75.7	88.4	88.4	**96.0**	**96.0**	94.3	94.3
061_foam_brick⋆	87.8	87.8	56.2	56.2	-	92.5	94.7	94.7	96.8	96.8	97.3	97.3	**98.5**	**98.5**
All Frames	60.0	75.9	57.5	72.8	-	93.1	83.8	92.6	91.8	95.5	**92.7**	**96.6**	87.9	96.5

Table 4. Refined and unrefined results on the *YCB-Video* dataset [18] with *ADD* and *ADD-S* area under curve scores in percent. For *PoseCNN* with multi-hypothesis *ICP*, results are taken from the corresponding publication [18]. To evaluate the refinement, predicted poses for *PoseCNN* [18] are taken from the *YCB_Video_toolbox*[1] while results for *Augmented Autoencoders*[2] [14] and *CosyPose*[3] [8] are computed using source code from the respective repositories.

Approach	PoseCNN [18]		PoseCNN[1] [18]				Augmented Autoencoders[2] [14]				CosyPose[3] [8]			
Refinement	MH ICP		-		ICG (Ours)		-		ICG (Ours)		Iterative Matching		IM + ICG (Ours)	
Objects	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S
002_master_chef_can	**69.0**	**95.8**	50.0	84.6	66.7	94.7	27.1	50.6	38.3	67.2	37.3	90.6	38.6	93.1
003_cracker_box	80.7	91.8	53.0	77.5	67.8	85.6	32.2	64.5	43.8	71.6	76.8	94.9	**81.4**	**97.9**
004_sugar_box	**97.2**	**98.2**	68.3	84.5	91.9	96.3	73.6	88.6	85.2	94.9	95.2	97.6	95.8	**98.2**
005_tomato_soup_can	81.6	94.5	66.1	81.4	82.8	91.3	72.3	84.4	82.3	90.0	90.5	94.6	**92.6**	**95.9**
006_mustard_bottle	**97.0**	**98.4**	80.8	91.1	93.9	97.4	77.5	90.9	87.9	96.9	92.7	96.5	96.3	**98.4**
007_tuna_fish_can	83.1	97.1	70.5	88.4	82.2	93.5	71.2	92.2	78.6	95.2	**93.9**	**97.5**	92.2	95.6
008_pudding_box	**96.6**	**97.9**	62.2	79.3	72.3	85.1	47.9	67.7	58.6	81.7	93.5	96.2	81.9	91.6
009_gelatin_box	**98.2**	**98.8**	74.9	87.7	95.1	97.8	74.8	82.9	81.9	88.9	94.1	96.1	93.5	97.9
010_potted_meat_can	**83.8**	**92.8**	59.3	78.8	69.1	82.4	53.6	63.3	61.7	68.0	75.9	84.0	78.8	85.7
011_banana	91.6	96.9	72.3	86.3	80.4	92.0	13.1	51.6	18.2	60.1	90.0	95.6	**95.2**	**98.2**
019_pitcher_base	**96.7**	97.8	52.9	77.6	85.9	93.6	77.6	91.7	92.1	98.0	94.0	97.3	**96.7**	**98.7**
021_bleach_cleanser	**92.3**	96.8	50.2	71.7	74.7	87.6	42.0	62.6	54.4	70.9	82.1	92.7	90.0	**97.8**
024_bowl	17.5	78.3	3.0	69.6	5.5	78.0	17.3	79.1	19.6	79.5	34.5	87.8	**36.6**	**89.8**
025_mug	81.4	95.1	58.4	78.8	88.2	96.6	58.0	80.9	82.8	93.6	87.8	94.9	**94.9**	**98.2**
035_power_drill	**96.9**	98.0	55.2	73.2	95.1	97.9	61.2	77.9	81.9	89.3	89.7	95.1	96.2	**98.3**
036_wood_block	**79.2**	**90.5**	26.4	64.3	35.5	69.9	1.6	55.2	2.5	60.8	24.8	80.5	29.0	87.4
037_scissors	**78.4**	**92.2**	34.8	55.9	59.0	79.6	0.8	7.0	0.7	7.5	67.6	81.5	73.9	86.9
040_large_marker	85.4	97.2	58.2	71.9	83.6	95.3	55.6	67.6	65.9	75.7	84.3	93.1	**90.8**	**97.5**
051_large_clamp	**52.6**	75.4	24.6	50.1	50.7	74.0	32.8	72.2	41.2	83.3	40.1	91.3	40.8	**94.6**
052_extra_large_clamp	28.7	65.3	16.3	44.5	25.8	67.7	26.9	59.5	32.5	63.6	40.2	**75.7**	**40.5**	75.1
061_foam_brick	48.3	97.1	40.4	88.2	42.2	92.5	19.4	56.2	22.5	57.7	51.7	94.7	**52.7**	**97.6**
All Frames	**79.3**	93.0	53.7	76.3	73.1	89.3	50.5	72.8	61.2	80.3	76.1	92.6	78.9	**94.7**

forming on par with the best reported results from *FFB6D*. This is remarkable since *FFB6D* trains on large amounts of pose-annotated real-world data that originates from a similar pose distribution. For many applications, such data is

not available. In contrast, *ICG* only requires a texture-less 3D model and no training data. With respect to the *ADD(S)* metric, both *PVN3D* and *FFB6D* report better results. The main reason for this is that our method is by design not considering texture and thus has a considerable disadvantage if the geometry is not conclusive. However, in return, there is no need for textured 3D models, which are required for all competing methods.

In contrast to 6DoF pose estimation methods, *ICG* considers the pose on a frame-to-frame basis without re-initialization. This leads to a small number of objects that get stuck in local minima and thus show relatively poor performance. On the other hand, *ICG* benefits from temporal consistency and performs more accurate than *FFB6D* for most objects. The advantage of our tracker becomes particularly obvious when comparing the required computation. While *FFB6D* depends on a high-end GPU and reports a runtime of $75\,\mathrm{ms}$ [2], *ICG* requires only $1.3\,\mathrm{ms}$ per frame on a single CPU core, which is $57\times$ faster. This is especially crucial in reactive, real-time applications for which hardware constraints exist. In conclusion, the experiment demonstrates that while tracking by detection is possible, for many real-world applications, it is not the most sensible solution. Given the high efficiency and good performance of *ICG*, in our opinion, it is best to rely on continuous 3D tracking for local pose updates while using 6DoF pose estimation for global initialization and long-term consistency.

6. 6-DoF Pose Refinement

Given that *ICG* is a local optimization method, the question emerges how well it would work for pose refinement. In the following, we thus use *ICG* to improve the predictions of *PoseCNN*, *Augmented Autoencoders*, and *CosyPose* and compare results on the *YCB-Video* dataset [18]. Depending on the pose estimation algorithm, errors along the principal axis are relatively large. To cope with those larger translational errors, we use the following parameters $r_\mathrm{t} = \{300, 250, 100\}$, $\sigma_\mathrm{d} = \{100, 50, 20\}$, $\lambda_\mathrm{t} = 100$, and conduct 7 instead of 4 iterations. For efficiency, strides are increased from $5\,\mathrm{mm}$ to $10\,\mathrm{mm}$. All other parameters remain the same as in the evaluation of tracking. With the increased number of iterations and considered area, the runtime increases to $2.7\,\mathrm{ms}$ per frame.

Results of the conducted evaluation are shown in Tab. 4. We thereby report both refined and unrefined scores for the considered 6DoF pose estimation methods. In addition, results from [18] are provided, which were obtained using an extensive multi-hypothesis *ICP* approach on the predictions of *PoseCNN*. According to [17], this refinement algorithm requires more than $10\,\mathrm{s}$ for a single pose. The evaluation shows that, even for the very good results of *CosyPose*, *ICG* is able to improve pose estimations for almost all objects. Also, it is interesting to see that, while it can not fully compete with extensive multi-hypothesis *ICP* refinement, the difference is not as big as one might expect. This is especially impressive considering that *ICG* is more than three orders of magnitude faster.

Table 5. Ablation study comparing refined results for *ICG* with and without the region modality to unrefined results. Values show *ADD* and *ADD-S* area under curve scores over all frames on the *YCB-Video* dataset [18] in percent.

Approach	PoseCNN[1] [18]		Augmented Autoencoders[2] [14]		CosyPose[3] [8]	
Refinement	ADD	ADD-S	ADD	ADD-S	ADD	ADD-S
Unrefined	53.7	76.3	50.5	72.8	76.1	92.6
ICG w/o Region	65.0	84.2	57.5	76.9	76.8	93.3
ICG w/ Region	73.1	89.3	61.2	80.3	78.9	94.7

Finally, we want to ensure that the pose refinement uses both depth and region information and that improvements are not only from the *ICP*-based depth modality. We thus conducted a short ablation study, for which results are shown in Tab. 5. The obtained scores demonstrate that *ICG* is not just a blown-up *ICP* approach but that the addition of region information significantly helps to improve performance. Given the good pose predictions and computational efficiency, we are thus confident that, while *ICG* is an excellent 3D object tracking approach, it also has many applications in pose refinement.

[1] https://github.com/yuxng/YCB_Video_toolbox
[2] https://github.com/DLR-RM/AugmentedAutoencoder
[3] https://github.com/ylabbe/cosypose

References

[1] Changhyun Choi and Henrik I. Christensen. RGB-D object tracking: A particle filter approach on GPU. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1084–1091, 2013. 1

[2] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. FFB6D: A full flow bidirectional fusion network for 6D pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3003–3013, 2021. 2, 3, 4

[3] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. PVN3D: a deep point-wise 3D keypoints voting network for 6DoF pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11629–11638, 2020. 2, 3

[4] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2020 on 6d object localization. In *Computer Vision – ECCV 2020 Workshops*, pages 577–594, 2020. 3

[5] Hong Huang, Fan Zhong, Yuqing Sun, and Xueying Qin. An occlusion-aware edge-based method for monocular 3d object tracking using edge confidence. *Computer Graphics Forum*, 39(7):399–409, 2020. 2

[6] Wadim Kehl, Federico Tombari, Slobodan Ilic, and Nassir Navab. Real-time 3D model tracking in color and depth on a single CPU core. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 465–473, 2017. 1

[7] Alexander Krull, Frank Michel, Eric Brachmann, Stefan Gumhold, Stephan Ihrke, and Carsten Rother. 6-DOF model based tracking via object coordinate regression. In *Asian Conference on Computer Vision*, pages 384–399, 2015. 1

[8] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. In *European Conference on Computer Vision*, pages 574–591, 2020. 2, 3, 4

[9] Jia-Chen Li, Fan Zhong, Song-Hua Xu, and Xue-Ying Qin. 3D object tracking with adaptively weighted local bundles. *Journal of Computer Science and Technology*, 36(3):555–571, 2021. 2

[10] Fulin Liu, Zhenzhong Wei, and Guangjun Zhang. An off-board vision system for relative attitude measurement of aircraft. *IEEE Transactions on Industrial Electronics*, 69(4):4225–4233, 2021. 2

[11] Manuel Stoiber, Martin Pfanne, Klaus H. Strobl, Rudolph Triebel, and Alin Albu-Schaeffer. A sparse gaussian approach to region-based 6DoF object tracking. In *Asian Conference on Computer Vision*, pages 666–682, 2020. 1

[12] Manuel Stoiber, Martin Pfanne, Klaus H. Strobl, Rudolph Triebel, and Alin Albu-Schaeffer. SRT3D: A sparse region-based 3D object tracking approach for the real world. *International Journal of Computer Vision*, 2022. 1, 2

[13] Xiaoliang Sun, Jiexin Zhou, Wenlong Zhang, Zi Wang, and Qifeng Yu. Robust monocular pose tracking of less-distinct objects based on contour-part model. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(11):4409–4421, 2021. 2

[14] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D orientation learning for 6D object detection from RGB images. In *European Conference on Computer Vision*, pages 712–729, 2018. 3, 4

[15] David J. Tan, Nassir Navab, and Federico Tombari. Looking beyond the simple scenarios: Combining learners and optimizers in 3D temporal tracking. *IEEE Transactions on Visualization and Computer Graphics*, 23(11):2399–2409, 2017. 1

[16] Henning Tjaden, Ulrich Schwanecke, Elmar Schömer, and Daniel Cremers. A region-based Gauss-Newton approach to real-time monocular multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1797–1812, 2018. 1, 2

[17] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. DenseFusion: 6D object pose estimation by iterative dense fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3338–3347, 2019. 3, 4

[18] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems*, 2018. 1, 2, 3, 4

[19] Leisheng Zhong, Xiaolin Zhao, Yu Zhang, Shunli Zhang, and Li Zhang. Occlusion-aware region-based 3D pose tracking of objects with temporally consistent polar-based local partitioning. *IEEE Transactions on Image Processing*, 29:5065–5078, 2020. 2