Deep Quaternion Pose Proposals for 6D Object Pose Tracking

Mateusz Majcher, Bogdan Kwolek

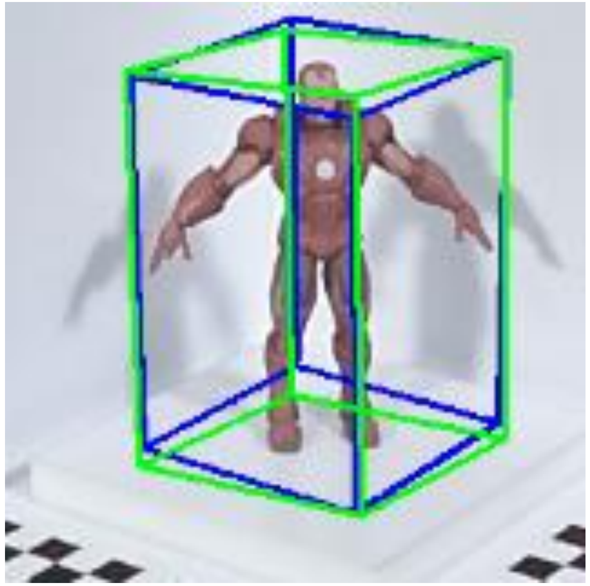
11 October, IWDSC 2021

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
AGH University of Science and Technology

Problem

Pose Tracking - Estimating the 6-DoF pose (3D rotations and 3D translations) of an object with respect to the camera with additional informations from previous frames.

Fig. 1: Examples of illumination differences and similarity of poses



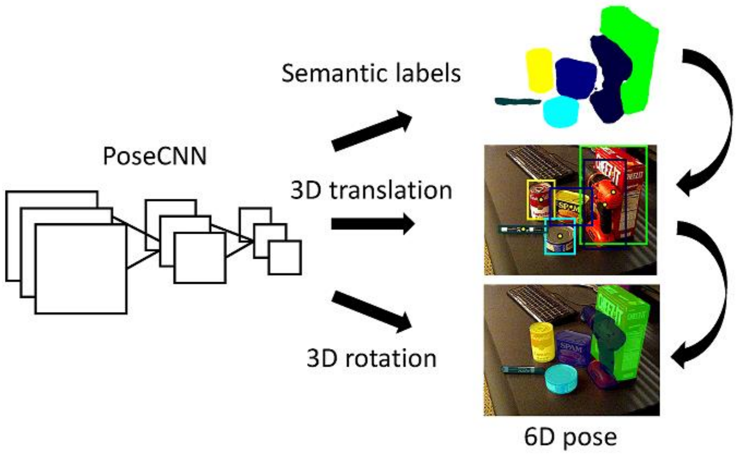
Motivation

Two main CNN-based approaches:

Motivation

Two main CNN-based approaches:

Direct 6D pose regression

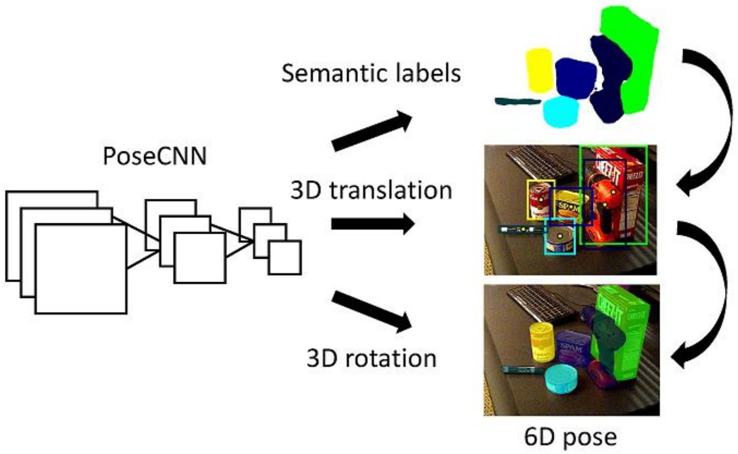


[Xiang et al, 2018]

Motivation

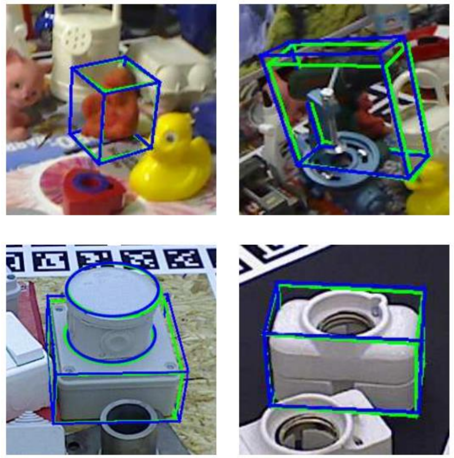
Two main CNN-based approaches:

Direct 6D pose regression



[Xiang et al, 2018]

Predicting 2D key-point locations

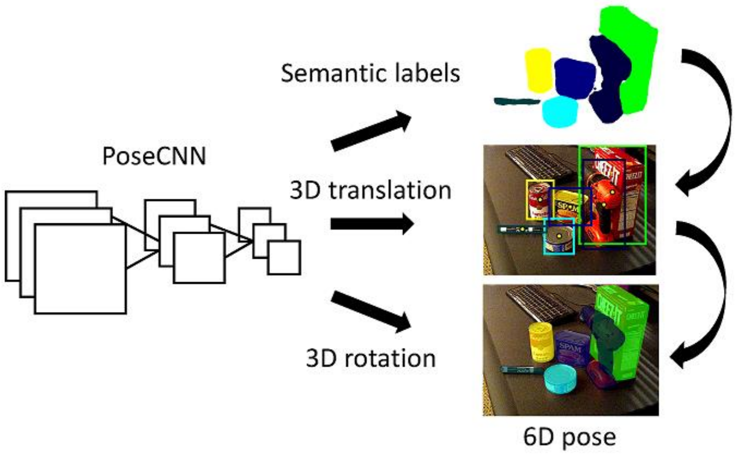


[Rad et al, 2017]

Motivation

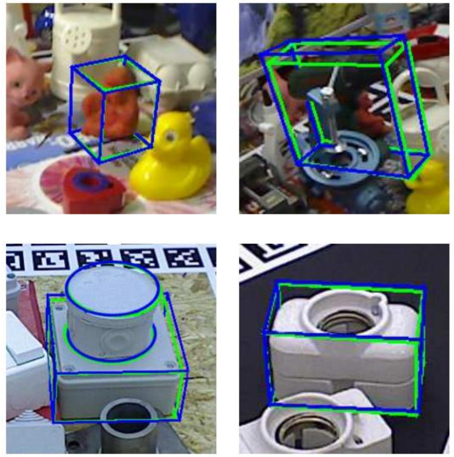
Two main CNN-based approaches:

Direct 6D pose regression



[Xiang et al, 2018]

Predicting 2D key-point locations



[Rad et al, 2017]

Most existing methods output a single guess of object's pose – we want a distribution of object poses.

Our approach

- We present a 6D object pose tracker, which uses quaternion pose distributions.

- We propose an unit quaternion representation of rotational state space for a particle filter and particle swarm optimization.

- A common Siamese neural network is employed to guide the particles.

- Our algorithm was evaluated on Nvidia Jetson AGX Xavier.

Quaternion

Represented as numbers with one real part and three distinct imaginary parts:

$$\boldsymbol{q} = q_w + q_x i + q_y j + q_z k$$

where q_w, q_x, q_y, q_z are real numbers and i, j, k satisfy:

$$i^2 = j^2 = k^2 = ijk = -1$$

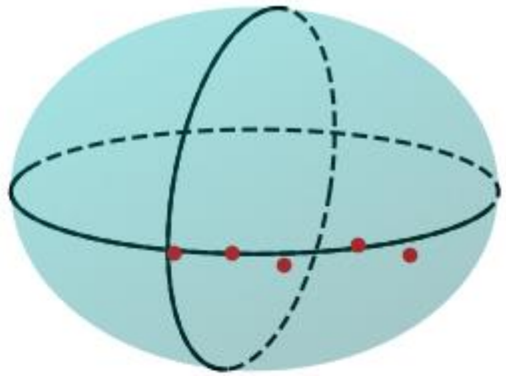


Fig. 1: Quaternion visualization

Quaternion Particle Filter

- Monte Carlo method
- Unit quaternion representation of rotation state space
- Translation vector in Euclidean space
- Weights are calculated on the basis of a probabilistic observation model

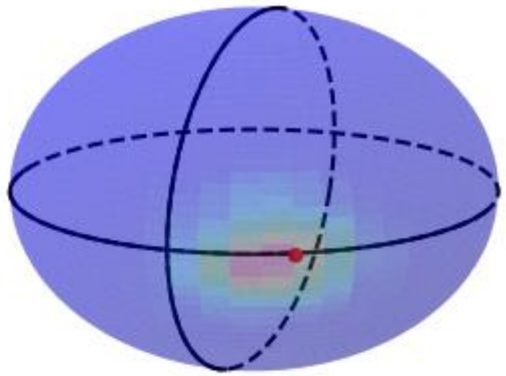


Fig. 2: Prior distribution

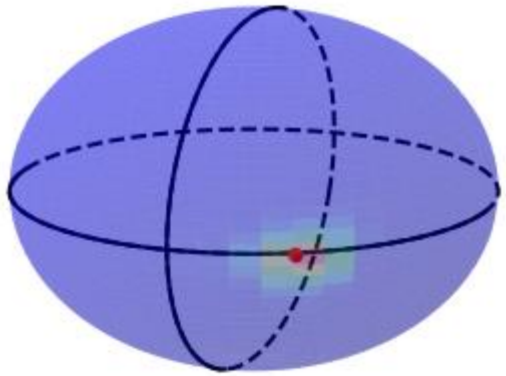


Fig. 3: Distribution after resampling

Quaternion Particle Swarm Optimization

- Population-based optimization method
- Iteratively tries to improve a candidate solution with respect to a fitness measure
- Particles are clustered using k-means++ algorithm

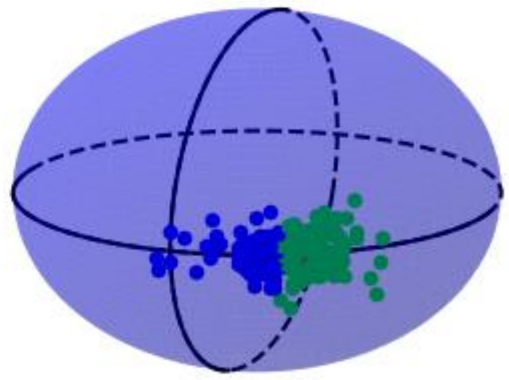


Fig. 4: K-means clusters

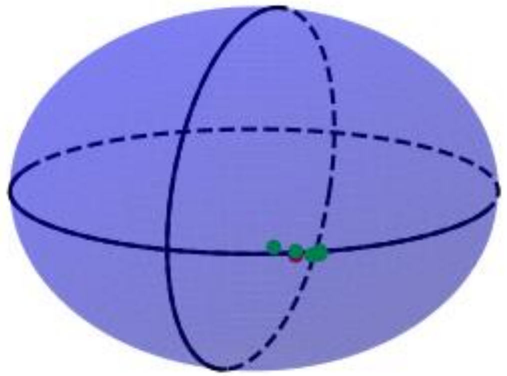


Fig. 5: After execution of PSO

Siamese Neural Network

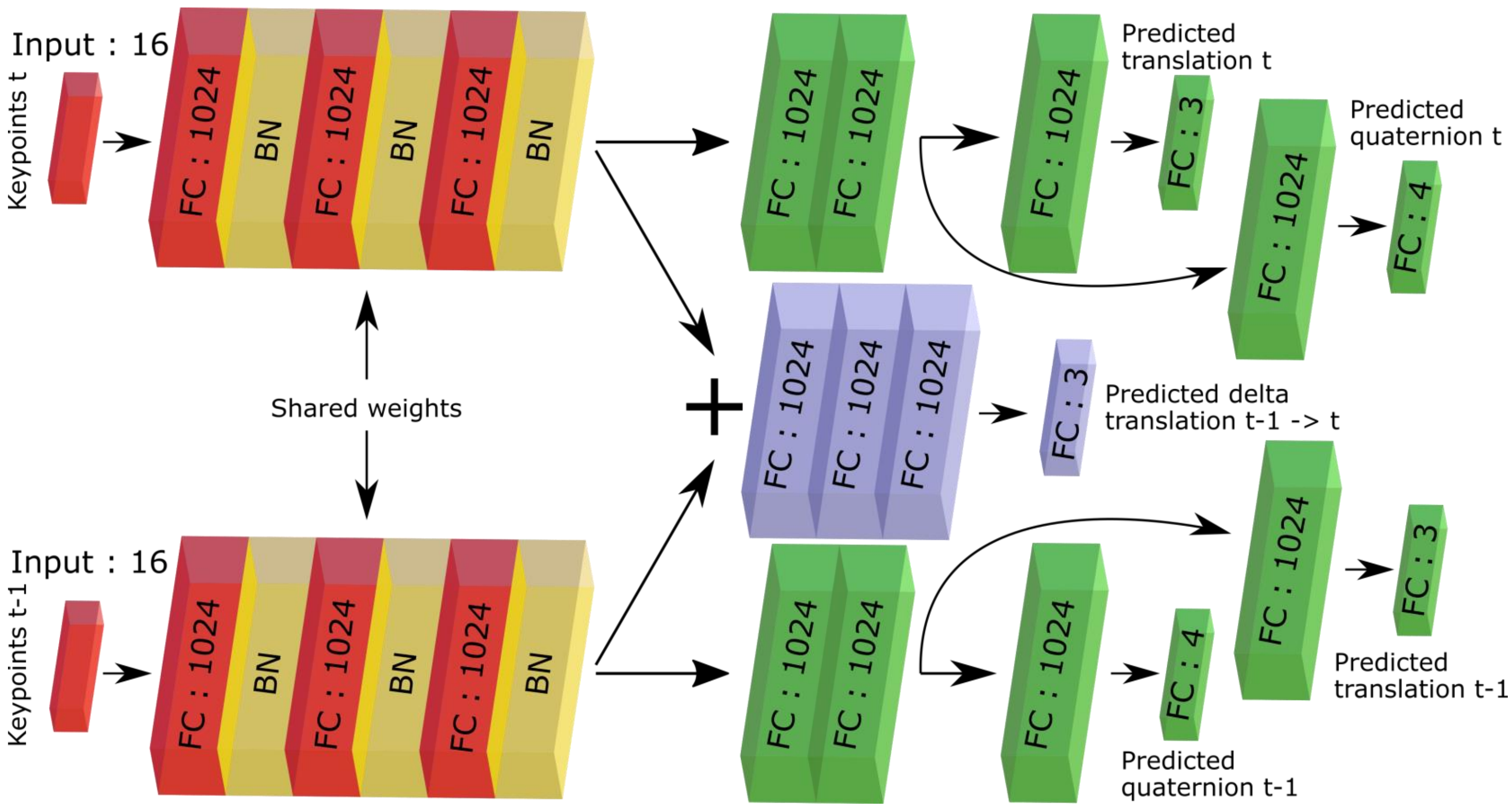
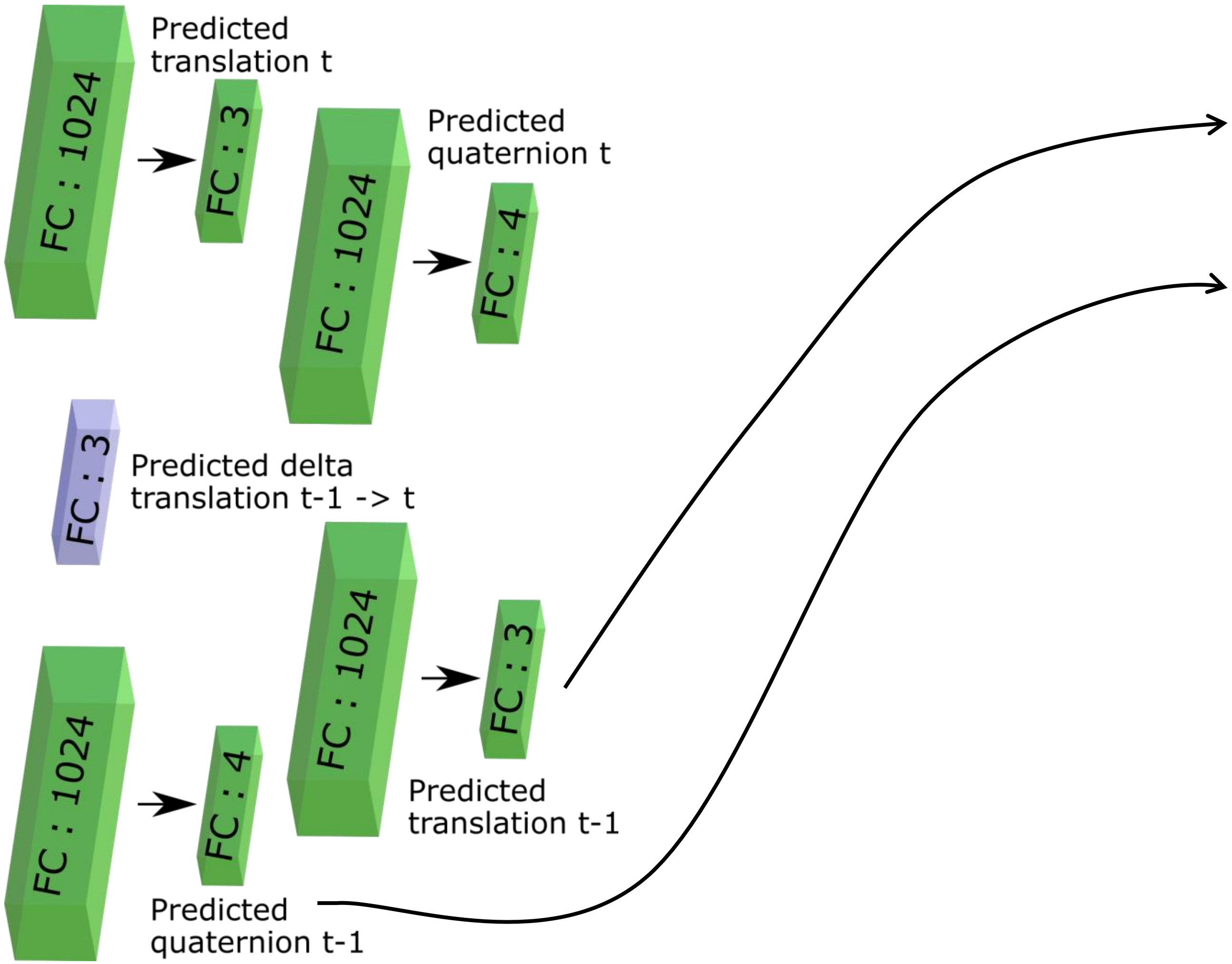


Fig. 6: Architecture of Siamese neural network for prediction of 6D pose of the object

Loss Function

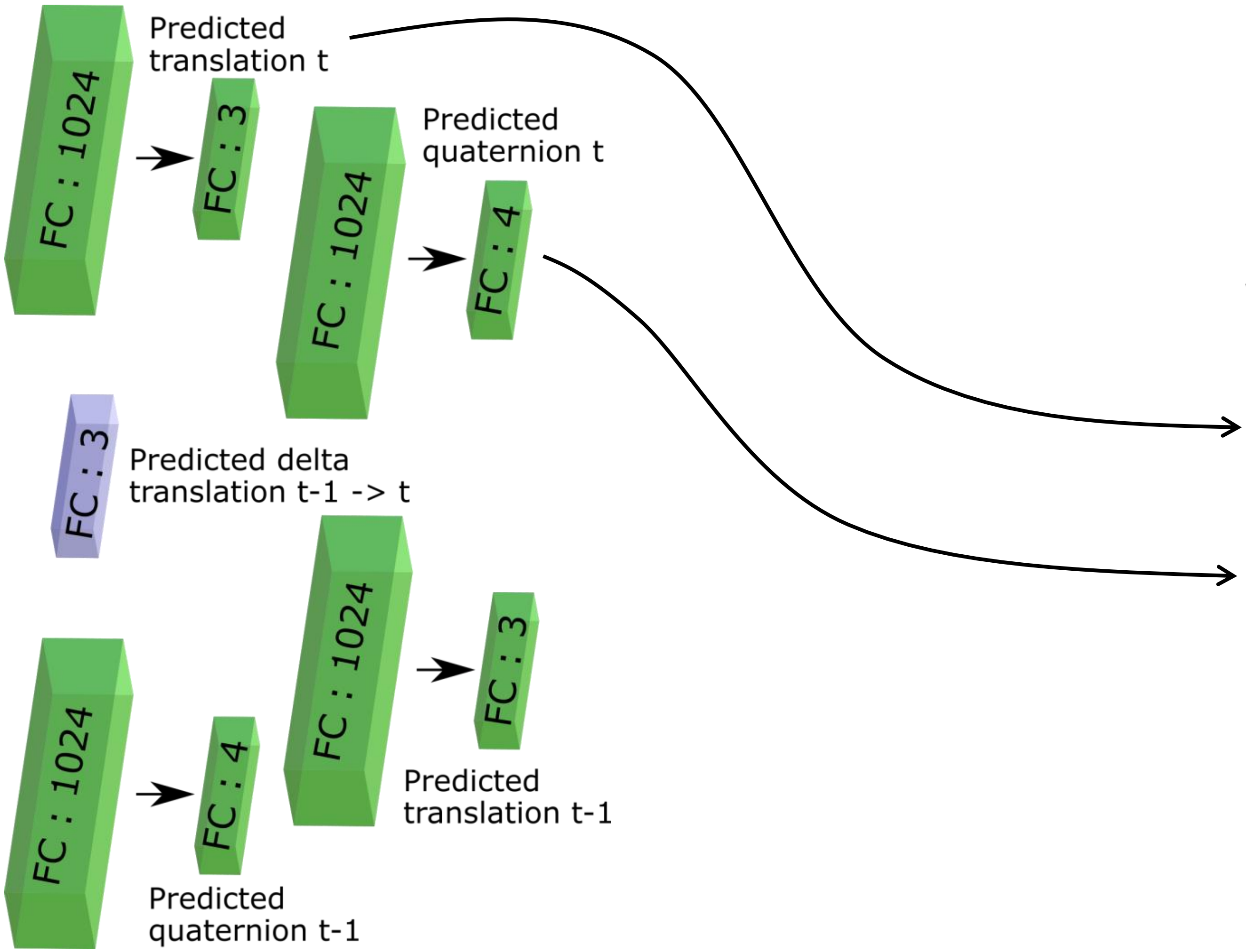
The loss function is sum of the following components:



$$L = L_{T,t-1} + L_{Q,t-1} + L_{T,t} + L_{Q,t} + L_{DT1} + L_{DT2}$$

Loss Function

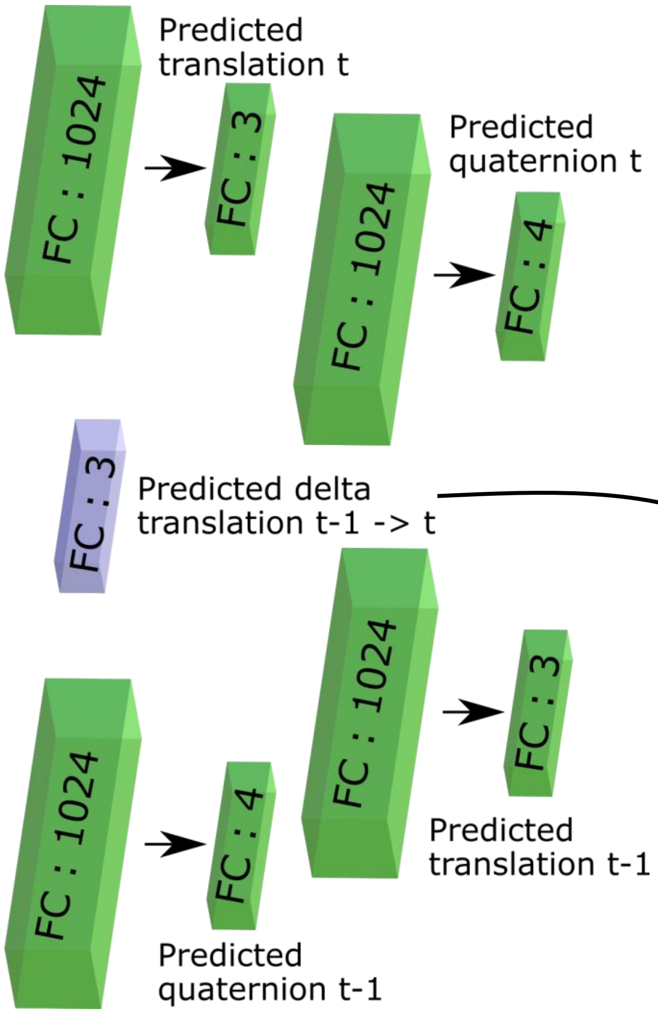
The loss function is sum of the following components:



$$L = L_{T,t-1} + L_{Q,t-1} + L_{T,t} + L_{Q,t} + L_{DT1} + L_{DT2}$$

Loss Function

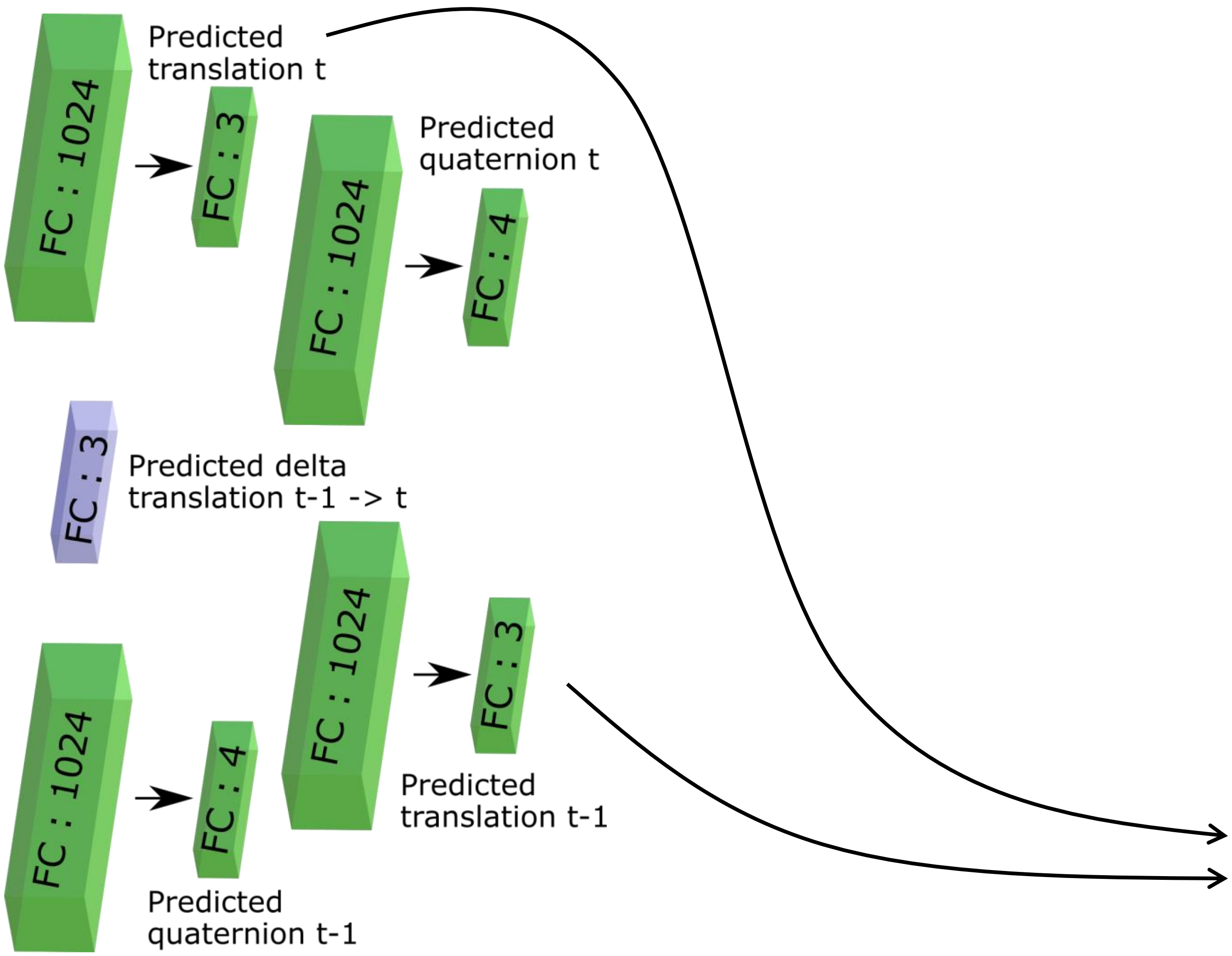
The loss function is sum of the following components:



$$L = L_{T,t-1} + L_{Q,t-1} + L_{T,t} + L_{Q,t} + L_{DT1} + L_{DT2}$$

Loss Function

The loss function is sum of the following components:



$$L = L_{T,t-1} + L_{Q,t-1} + L_{T,t} + L_{Q,t} + L_{DT1} + L_{DT2}$$

Algorithm

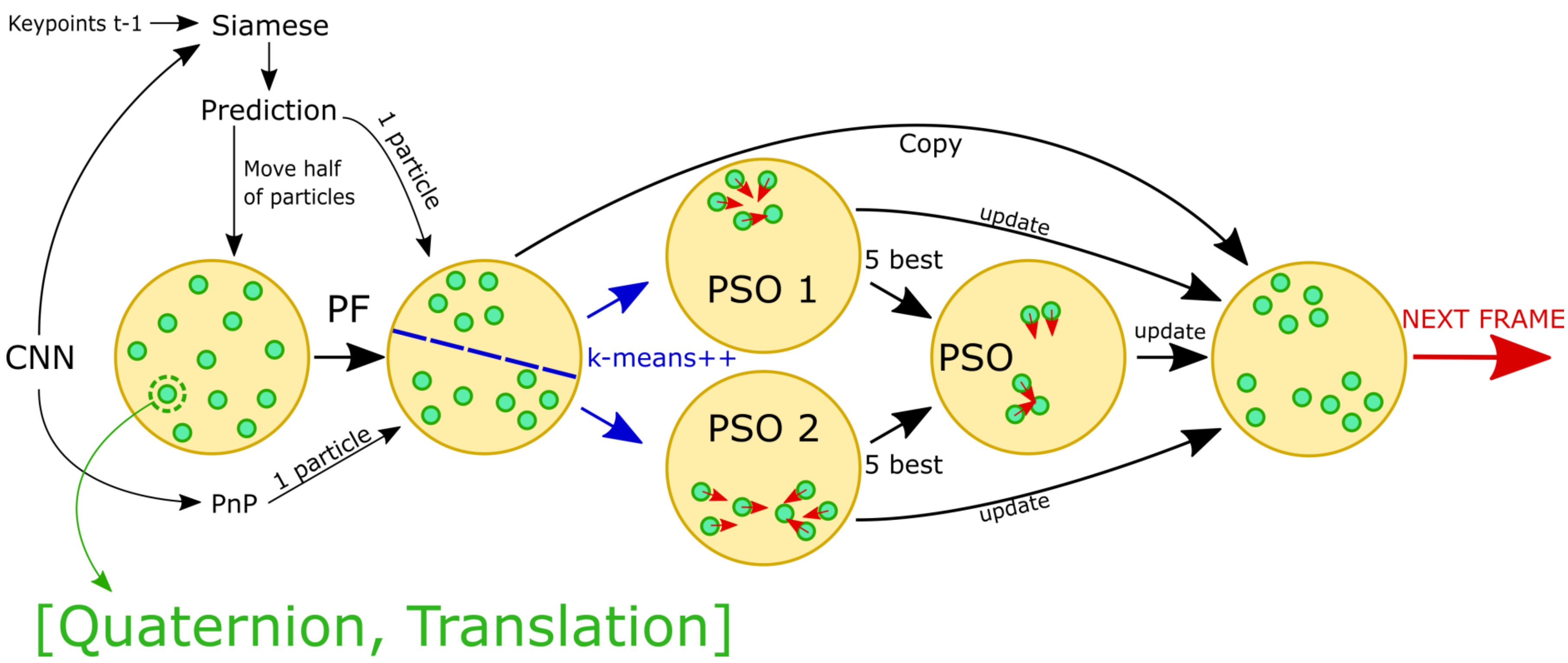


Fig. 7: Q-PF-PSO supported with Siamese neural network-based object pose predictions

Algorithm

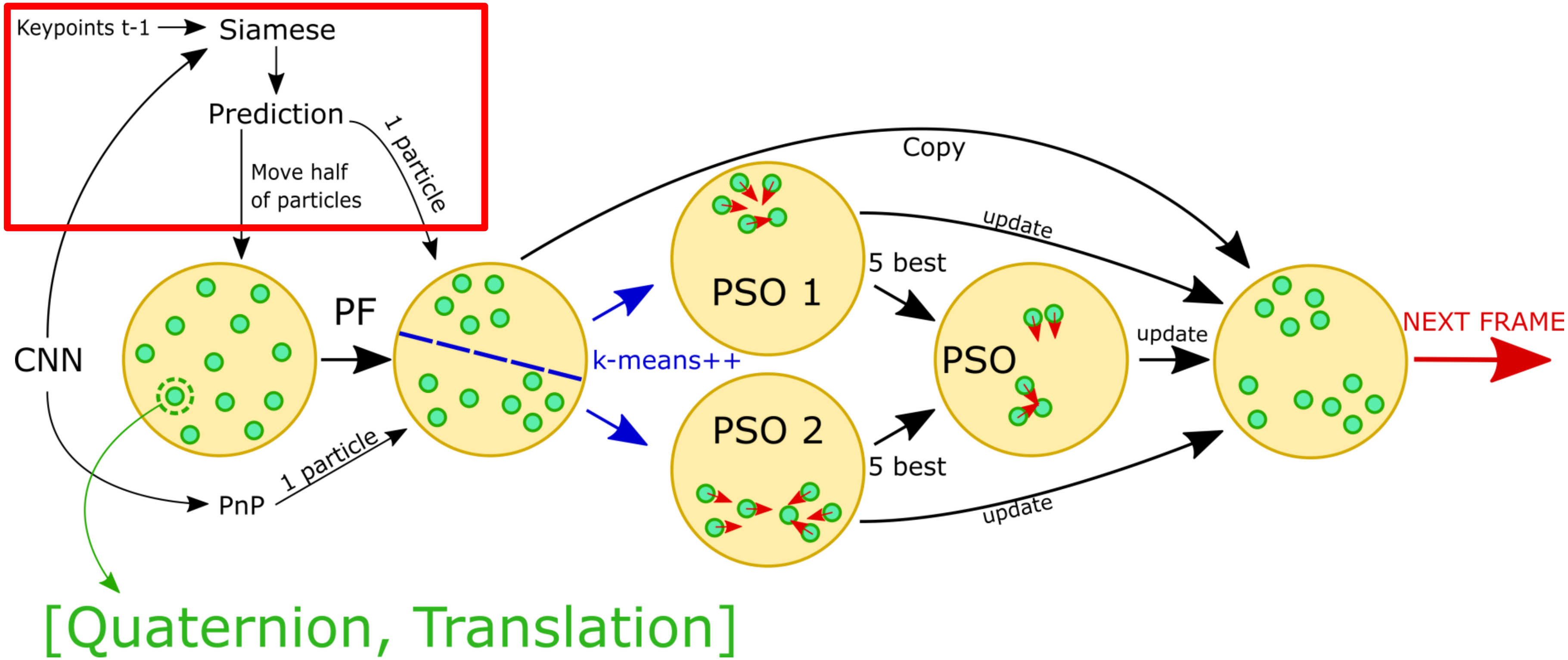


Fig. 7: Q-PF-PSO supported with Siamese neural network-based object pose predictions

Algorithm

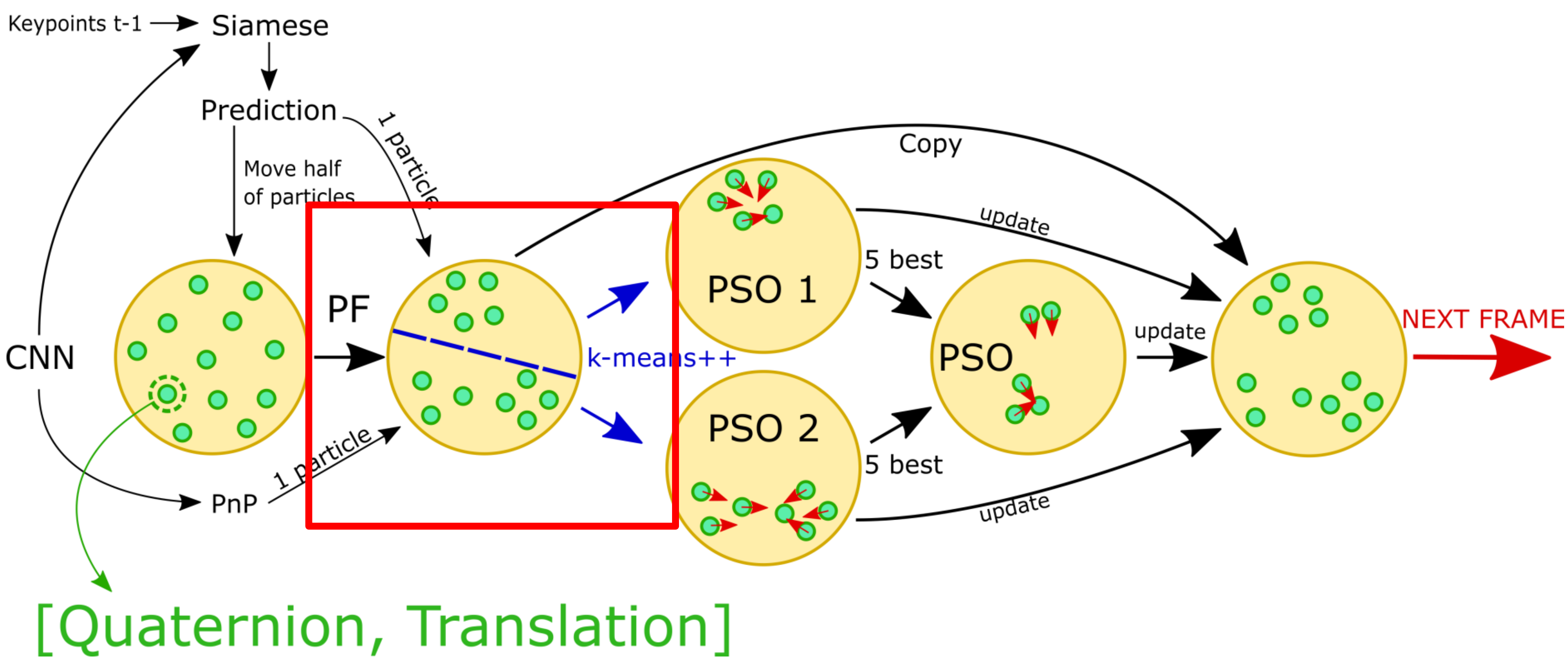


Fig. 7: Q-PF-PSO supported with Siamese neural network-based object pose predictions

Algorithm

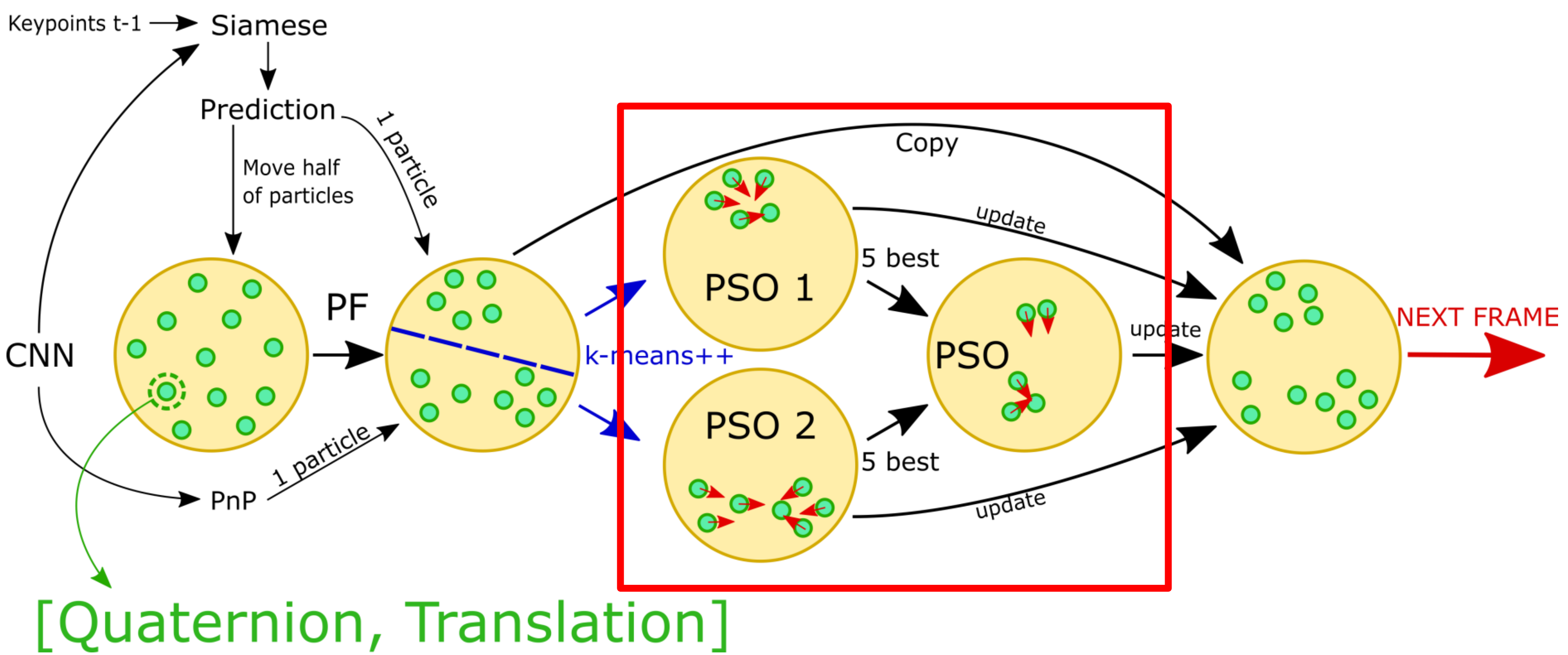


Fig. 7: Q-PF-PSO supported with Siamese neural network-based object pose predictions

Experimental Results

We evaluated our algorithm on freely available OPT benchmark dataset and our own dataset.

view, ADD [%]	House (nS)	House	Ironman (nS)	Ironman
Behind, 10%	85±2.07	82±2.53	64±2.29	77±3.06
Behind, 20%	99±0.51	97±1.68	91±1.62	95±2.06
Left, 10%	81±1.52	76±2.58	35±5.56	42±3.37
Left, 20%	97±1.34	98±1.34	66±6.77	69±3.82
Right, 10%	55±5.13	75±2.02	46±5.15	56±4.34
Right, 20%	74±7.20	96±1.80	73±6.73	79±5.04
Front, 10%	53±3.66	82±1.14	55±6.02	68±3.45
Front, 20%	81±5.40	97±1.10	73±5.94	83±3.29
Average, 10%	68	79	50	61
Average, 20%	88	97	76	82

Table 1: Tracking scores (nS – no Siamese)

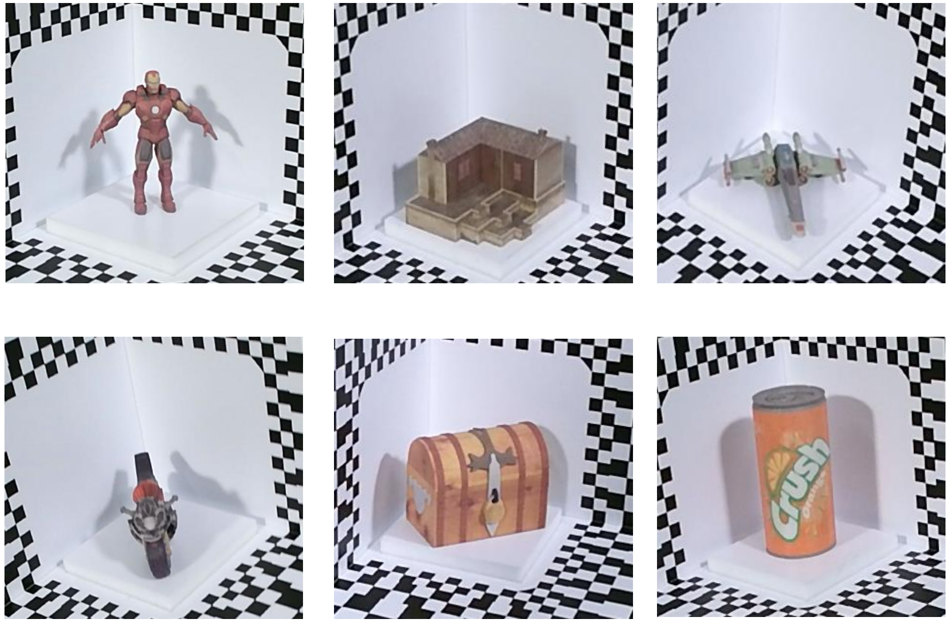


Fig. 9: Objects from OPT dataset

AUC score [%]	House	Ironman	Jet	Bike	Chest	Soda
PWP3D	3.58	3.92	5.81	5.36	5.55	5.87
UDP	5.97	5.25	2.34	6.10	6.79	8.49
ElasticFusion	2.70	1.69	1.86	1.57	1.53	1.90
Reg. G-N.	10.15	11.99	13.22	11.90	11.76	8.86
w/o Siamese	11.52	9.26	9.19	10.81	6.93	6.61
Siamese for each	**13.68**	10.59	10.37	**12.36**	7.80	8.60
Siamese for all	13.27	10.32	10.33	11.88	7.60	**8.90**

Table 2: AUC scores in FreeMotion scenario

Experimental Results

We evaluated our algorithm on freely available OPT benchmark dataset and our own dataset.

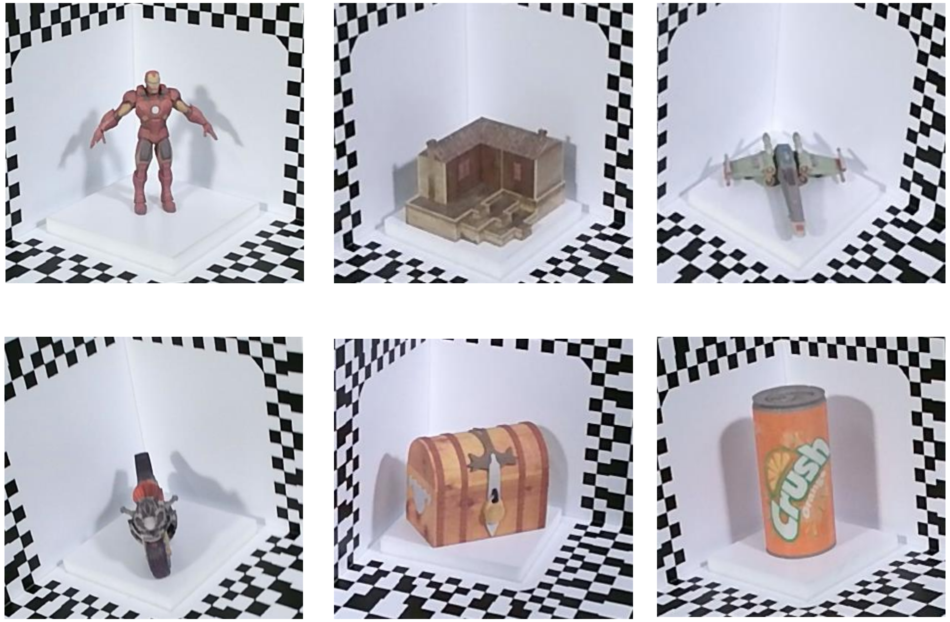


Fig. 9: Objects from OPT dataset

view, ADD [%]	House (nS)	House	Ironman (nS)	Ironman
Behind, 10%	85±2.07	82±2.53	64±2.29	77±3.06
Behind, 20%	99±0.51	97±1.68	91±1.62	95±2.06
Left, 10%	81±1.52	76±2.58	35±5.56	42±3.37
Left, 20%	97±1.34	98±1.34	66±6.77	69±3.82
Right, 10%	55±5.13	75±2.02	46±5.15	56±4.34
Right, 20%	74±7.20	96±1.80	73±6.73	79±5.04
Front, 10%	53±3.66	82±1.14	55±6.02	68±3.45
Front, 20%	81±5.40	97±1.10	73±5.94	83±3.29
Average, 10%	68	79	50	61
Average, 20%	88	97	76	82

Table 1: Tracking scores (nS – no Siamese)

AUC score [%]	House	Ironman	Jet	Bike	Chest	Soda
PWP3D	3.58	3.92	5.81	5.36	5.55	5.87
UDP	5.97	5.25	2.34	6.10	6.79	8.49
ElasticFusion	2.70	1.69	1.86	1.57	1.53	1.90
Reg. G-N.	10.15	11.99	13.22	11.90	11.76	8.86
w/o Siamese	11.52	9.26	9.19	10.81	6.93	6.61
Siamese for each	**13.68**	10.59	10.37	**12.36**	7.80	8.60
Siamese for all	13.27	10.32	10.33	11.88	7.60	**8.90**

Table 2: AUC scores in FreeMotion scenario

Experimental Results

We evaluated our algorithm on freely available OPT benchmark dataset and our own dataset.

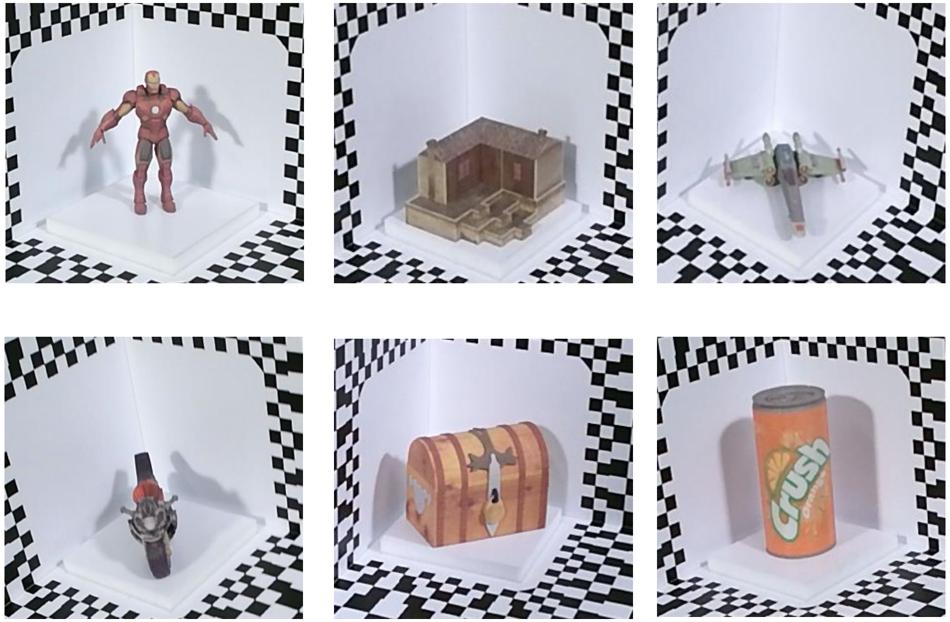


Fig. 9: Objects from OPT dataset

view, ADD [%]	House (nS)	House	Ironman (nS)	Ironman
Behind, 10%	85±2.07	82±2.53	64±2.29	77±3.06
Behind, 20%	99±0.51	97±1.68	91±1.62	95±2.06
Left, 10%	81±1.52	76±2.58	35±5.56	42±3.37
Left, 20%	97±1.34	98±1.34	66±6.77	69±3.82
Right, 10%	55±5.13	75±2.02	46±5.15	56±4.34
Right, 20%	74±7.20	96±1.80	73±6.73	79±5.04
Front, 10%	53±3.66	82±1.14	55±6.02	68±3.45
Front, 20%	81±5.40	97±1.10	73±5.94	83±3.29
Average, 10%	68	79	50	61
Average, 20%	88	97	76	82

Table 1: Tracking scores (nS – no Siamese)

AUC score [%]	House	Ironman	Jet	Bike	Chest	Soda
PWP3D	3.58	3.92	5.81	5.36	5.55	5.87
UDP	5.97	5.25	2.34	6.10	6.79	8.49
ElasticFusion	2.70	1.69	1.86	1.57	1.53	1.90
Reg. G-N.	10.15	11.99	13.22	11.90	11.76	8.86
w/o Siamese	11.52	9.26	9.19	10.81	6.93	6.61
Siamese for each	**13.68**	10.59	10.37	**12.36**	7.80	8.60
Siamese for all	13.27	10.32	10.33	11.88	7.60	**8.90**

Table 2: AUC scores in FreeMotion scenario

Experimental Results

Fig. 10: Objects from our dataset

	PC	Jetson
U-Net	0.040	0.020
Keypoints	0.035	0.040
Siamese	0.030	0.007
k-means++	0.005	0.010
PSO 200p. 3 iter.	0.037	0.060
PSO 10p. 10 iter.	0.026	0.040
overheads	0.017	0.023
Total	0.190	0.200

Table 3: Running times

tracking score [%]	Avg., ADD 10%	Avg, ADD 20%
drill w/o Siam.	80	97
drill with Siam. sep.	83	96
drill with Siam. com	**84**	94
frog w/o Siam.	65	79
frog with Siam. sep.	**76**	86
frog with Siam. com.	75	88
pig w/o Siam.	61	81
pig with Siam. sep.	**68**	80
pig with Siam. com.	66	82
duck w/o Siam.	72	90
duck with Siam. sep.	75	90
duck with Siam. com.	**77**	90
ext. w/o Siam.	54	70
ext. with Siam. sep.	**63**	76
ext. with Siam. com.	62	74
mult. w/o Siam.	67	79
mult. with Siam. sep.	**69**	80
mult. with Siam. com.	68	81

Table 4: ADD scores on our dataset

Experimental Results

Fig. 10: Objects from our dataset

	PC	Jetson
U-Net	0.040	0.020
Keypoints	0.035	0.040
Siamese	0.030	0.007
k-means++	0.005	0.010
PSO 200p. 3 iter.	0.037	0.060
PSO 10p. 10 iter.	0.026	0.040
overheads	0.017	0.023
Total	0.190	0.200

Table 3: Running times

tracking score [%]	Avg., ADD 10%	Avg, ADD 20%
drill w/o Siam.	80	97
drill with Siam. sep.	83	96
drill with Siam. com	**84**	94
frog w/o Siam.	65	79
frog with Siam. sep.	76	86
frog with Siam. com.	75	88
pig w/o Siam.	61	81
pig with Siam. sep.	**68**	80
pig with Siam. com.	66	82
duck w/o Siam.	72	90
duck with Siam. sep.	75	90
duck with Siam. com.	**77**	90
ext. w/o Siam.	54	70
ext. with Siam. sep.	63	76
ext. with Siam. com.	62	74
mult. w/o Siam.	67	79
mult. with Siam. sep.	**69**	80
mult. with Siam. com.	68	81

Table 4: ADD scores on our dataset

Experimental Results

Fig. 10: Objects from our dataset

	PC	Jetson
U-Net	0.040	0.020
Keypoints	0.035	0.040
Siamese	0.030	0.007
k-means++	0.005	0.010
PSO 200p. 3 iter.	0.037	0.060
PSO 10p. 10 iter.	0.026	0.040
overheads	0.017	0.023
Total	0.190	0.200

Table 3: Running times

tracking score [%]	Avg., ADD 10%	Avg, ADD 20%
drill w/o Siam.	80	97
drill with Siam. sep.	83	96
drill with Siam. com	84	94
frog w/o Siam.	65	79
frog with Siam. sep.	76	86
frog with Siam. com.	75	88
pig w/o Siam.	61	81
pig with Siam. sep.	68	80
pig with Siam. com.	66	82
duck w/o Siam.	72	90
duck with Siam. sep.	75	90
duck with Siam. com.	77	90
ext. w/o Siam.	54	70
ext. with Siam. sep.	63	76
ext. with Siam. com.	62	74
mult. w/o Siam.	67	79
mult. with Siam. sep.	69	80
mult. with Siam. com.	68	81

Table 4: ADD scores on our dataset

Conclusion

- We have presented that Siamese neural network can deliver pose predictions over time, which improve the performance of the object tracking.
- We proposed an unit quaterion representation of the rotational state space for particle filter hybridized with particle swarm optimization.
- Our algorithm delivers probability distribution of object poses in contrast to recent approaches.
- Our system has been evalueted on Nvidia Jetson with resonable running time.

Deep Quaternion Pose Proposals for 6D Object Pose Tracking

Mateusz Majcher, Bogdan Kwolek

majcher@agh.edu.pl bkw@agh.edu.pl

11 October, IWDSC 2021

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
AGH University of Science and Technology