FAST FEATURE-LESS QUATERNION BASED PARTICLE SWARM OPTIMISATION FOR OBJECT POSE ESIMATION FOR RGB-D IMAGES

Giorgio Toscana
Politecnico di Torino
giorgio.toscana@polito.it

POLITECNICO DI TORINO

ABSTRACT

We present a novel quaternion-based formulation of Particle Swarm Optimization for object pose estimation which, differently from other approaches, does not rely on 2D or 3D features or machine learning.
We are interested in pose estimation of objects with large variety in appearance, from lack of texture to strong textures, for the task of robotic grasping.
The quaternion formulation avoids the gimbal lock problem and, unlike other rotation formalisms, doesn't require conversions to and from rotation matrix form at each step. The objective function is based on raw 2D depth information only, under the assumption that the object region is segmented from the background.
The algorithm is implemented on GPU, and the nature of the objective function allows high parallelization.

METHOD

Why Unitary Quaternions?

$$\mathbf{q} = [q_0,\ \vec{q}\,]; \quad q_0 \in \mathbb{R}, \quad \vec{q} \in \mathbb{R}^3$$
$$\|\mathbf{q}\| = 1 \qquad (1)$$

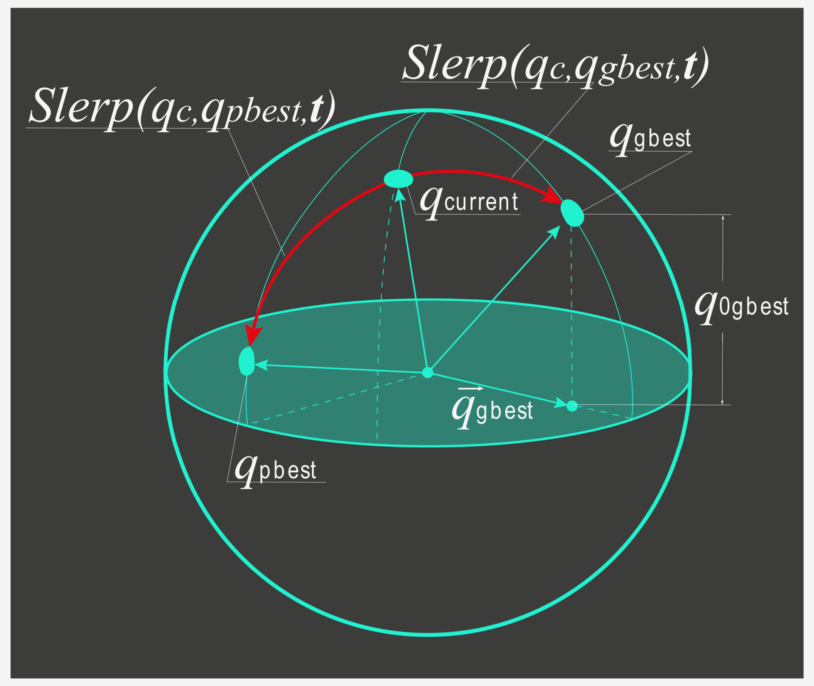
Eq. (1) is the equation of the **hypersphere** $\mathbb{S}^3$ embedded in a 4D Euclidean space. Hence any unit quaternion **q** lies on the surface of $\mathbb{S}^3$

Motivations

- Gimbal-lock free
- Well defined SLERP interpolation
- No conversion to and from rotation matrix form required at each iteration.

PSO Angular velocity and orientation update

The goal is to obtain both the **cognitive** and **social angular velocity** effecting an object, through the **quaternion inverse displacement**.
Let **q**0, **q**1 $\in \mathbb{S}^3$ and $t \in \mathbb{R}$ with $0 \leq t \leq 1$; the SLERP and its derivative are defined as:



$$\mathbf{q}(t) = (\mathbf{q}_1 \star \mathbf{q}_0^*)^t \star \mathbf{q}_0 = Slerp(\mathbf{q}_0, \mathbf{q}_1, t)$$
$$\dot{\mathbf{q}}(t) = \mathrm{Log}(\mathbf{q}_1 \star \mathbf{q}_0^*) \star \mathbf{q}(t) = \frac{\mathrm{d}Slerp}{\mathrm{d}t}$$
$$\dot{\mathbf{q}}(t) = \frac{1}{2}\mathbf{q}(t) \star \boldsymbol{\omega}(t)$$

The instantaneous angular velocity needed to rotate the object from the initial orientation (**q**0) to the final one (**q**1) is:

$$\boldsymbol{\omega}(t) = 2\mathrm{Log}\left(\mathbf{q}_1 \star \mathbf{q}_0^*\right)$$

The angular velocity update equation for the i-th particle is formulated as follows:

$$\begin{aligned}\boldsymbol{\omega}_i(t+1) = \quad & w\boldsymbol{\omega}_i(t) + \\ + \quad & c_1 r_1 \left[2\mathrm{Log}\left(\mathbf{q}_{pbest_i}(t) \star \mathbf{q}^*_{current_i}(t)\right)\right] + \\ + \quad & c_2 r_2 \left[2\mathrm{Log}\left(\mathbf{q}_{gbest}(t) \star \mathbf{q}^*_{current_i}(t)\right)\right]\end{aligned}$$

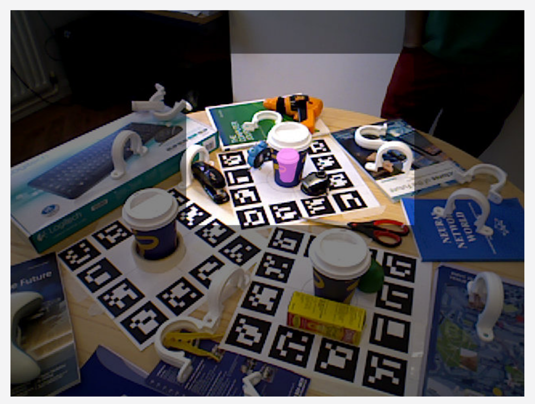
The orientation of the i-th particle is then updated by means of the discrete form of the quaternion kinematics:

$$\mathbf{q}_i(t+1) = \cos(\psi(t))\,\mathbf{q}_i(t) + \frac{1}{2}\frac{\sin(\psi(t))}{\psi(t)}\mathbf{q}_i(t) \star \boldsymbol{\omega}_i(t+1)T_c,$$
$$\psi(t) = \|\boldsymbol{\omega}_i(t+1)\|_2 \frac{T_c}{2}$$

Fitness Function

Each particle renders its pose hypothesis against the depth map of the cluster. The fitness value of the j-th particle is thus computed as follows:

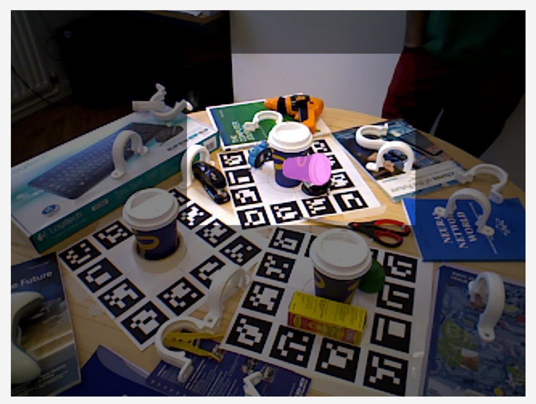
$$\Phi_j = \frac{\alpha}{N_{R_j}} \sum_{i=1}^{N_{R_j}} \left(z_{K_i} - z_{R_{ij}}\right)^2 + \beta\ \frac{\mu_j + \kappa_j}{2}$$



- CASE 1
- Rendered Area = 36 px
- Cluster Area = 100 px

$\mu = \frac{100-36}{100} = 0.64$

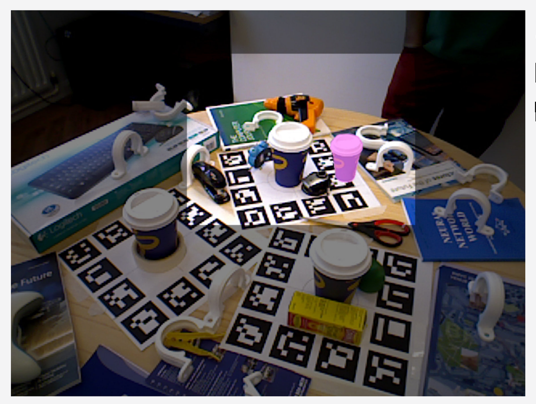
$\kappa = \frac{0}{36} = 0$



- CASE 2
- Rendered Area = 36 px
- Cluster Area = 100 px

$\mu = \frac{100-18}{100} = 0.82$

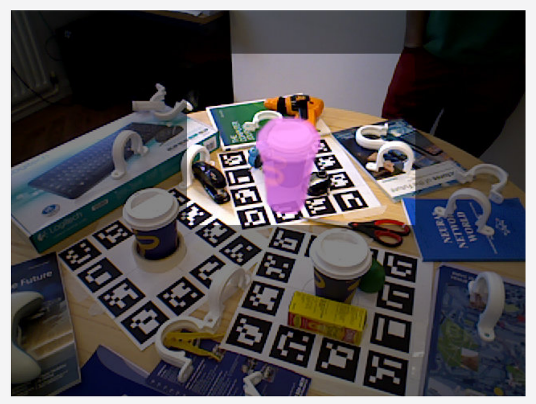
$\kappa = \frac{18}{36} = 0.5$



- CASE 3
- Rendered Area = 36 px
- Cluster Area = 100 px

$\mu = \frac{100-0}{100} = 1$

$\kappa = \frac{36}{36} = 1$



- CASE 4
- Rendered Area = 400 px
- Cluster Area = 100 px

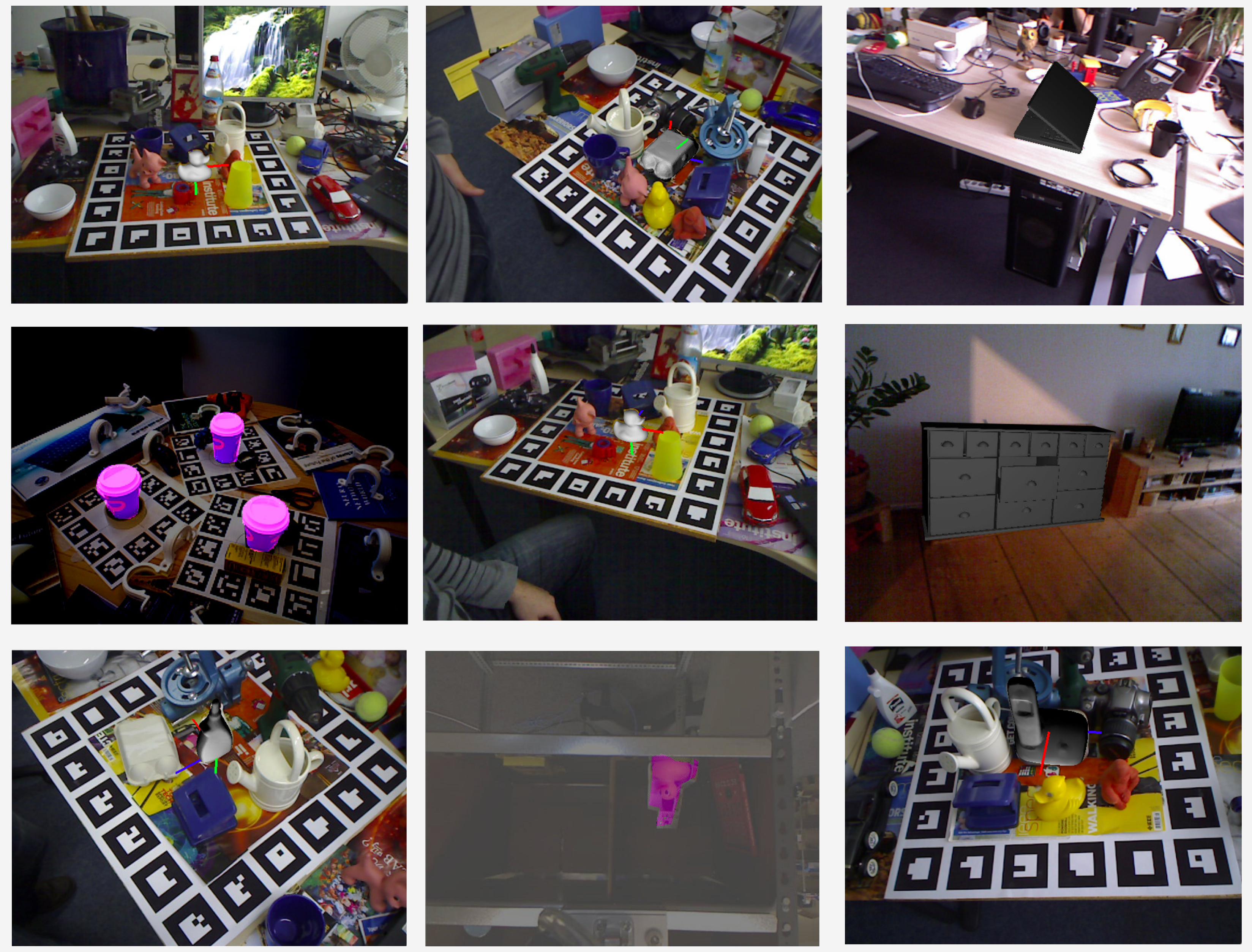
$\mu = \frac{0}{100} = 0$

$\kappa = \frac{400-100}{400} = 0.75$

RESULTS

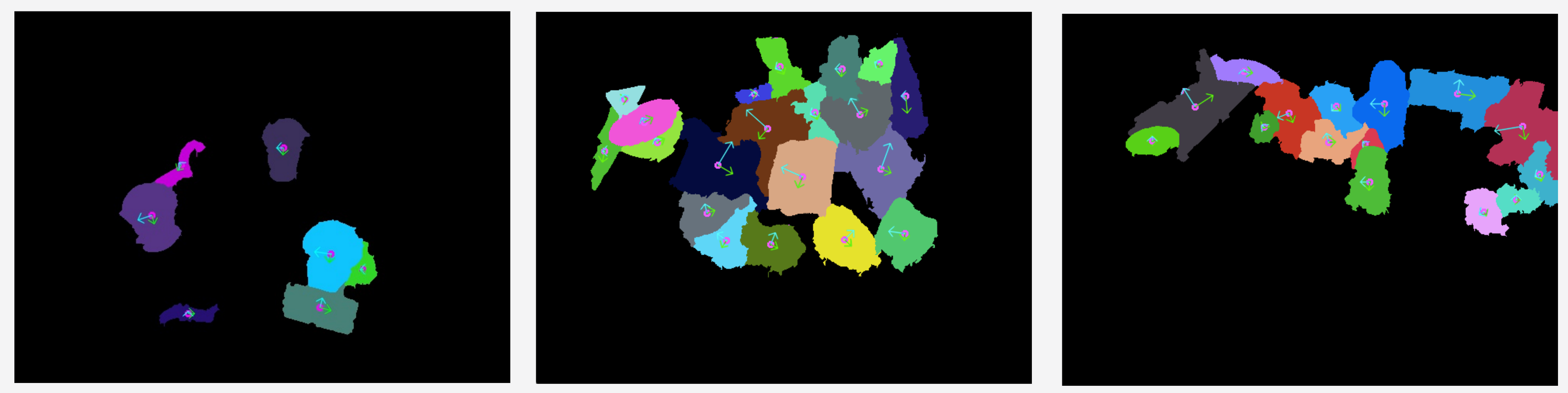
Rigid and Articulated Objects

We tested our approach on three public datasets [1], [2] and [3].
We used 1024 particles for the PSO and run 10 PSO iterations for each segmented cluster. Total time is 85ms for each cluster.



Segmentation Phase

Our Graph Based Segmentation Algorithm used to define the objects cluster in a 2D image.
The clusters are then employed as input of the Pose Estimation Algorithm.



Approach / Sequence	[1]	[2]	Our Approach
Bench Vise	0.85	0.96	0.72
Driller	0.69	0.9	0.9
Phone	0.56	0.73	0.98
Duck	0.58	0.91	0.97
Eggbox	0.86	0.74	0.95
Glue	0.44	0.68	0.8

Approach / Sequence	[1]	[2]	Our Approach
Coffee Cup	0.82	0.88	0.85
Shampoo	0.63	0.76	0.9
Juice Carton	0.49	0.87	0.4
Milk	0.18	0.39	0.67

References

[1] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. 2011.

[2] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class hough forests for 3d object detection and pose estimation. In Computer Vision–ECCV 2014, pages 462–477. Springer, 2014.

[3] E.B.M.Y.Y.S.G.C.R.FrankMichel, AlexanderKrull, "Pose estimation of kinematic chain instances via object coordinate regression," in BMVC 2015, 2015.