



Dynamical Pose Estimation with Graduated Non-Convexity for Outlier Robustness

Torbjørn Smith¹ Olav Egeland¹

¹*Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway. E-mail: {torbjorn.smith, olav.egeland}@ntnu.no*

Abstract

In this paper we develop a method for relative pose estimation for two sets of corresponding geometric primitives in 3D with a significant outlier fraction. This is done by using dynamical pose estimation as a solver in registration problems formulated with graduated non-convexity for truncated least squares (GNC-TLS). Dynamical pose estimation provides a unifying solver that can be used for point cloud registration, primitive registration, and absolute pose estimation. The solver is straightforward to implement, and it does not require specialized software for optimization. The main contribution of this paper is to show how the dynamical pose estimation method can be extended to fit into the GNC-TLS framework so that high outlier fractions can be handled. The proposed method is validated for point cloud registration, primitive registration, and absolute pose estimation. The accuracy and robustness to outliers is shown to be on the level of existing GNC-TLS methods.

Keywords: Pose estimation, Outlier rejection, Dynamical Pose Estimation, Graduated Non-Convexity

1 Introduction

Pose estimation with a high outlier fraction is important in robotics, and is needed in 3D registration problems like point cloud registration, primitive registration, absolute pose estimation and category registration.

When there are no outliers the pose estimation problem can be formulated as the least-squares minimization problem

$$\min_{\mathbf{T} \in SE(3)} \sum_{i=1}^N r(\mathbf{T} \otimes \mathbf{X}_i, \mathbf{Y}_i)^2 \quad (1)$$

where \mathbf{X}_i and \mathbf{Y}_i are corresponding geometric primitives, $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in SE(3)$ is a rigid transformation, $\mathbf{T} \otimes \mathbf{X}_i$ is the rigid transformation of \mathbf{X}_i , and the residual error $r(\mathbf{T} \otimes \mathbf{X}_i, \mathbf{Y}_i)$ is the distance from $\mathbf{T} \otimes \mathbf{X}_i$ to \mathbf{Y}_i . In point cloud registration this problem is solved in closed form in (Arun et al., 1987) and (Horn, 1987). Primitive registration with point-

to-plane, point-to-line and point-to-point correspondences was solved with branch and bound in (Hartley and Kahl, 2009) and (Olsson et al., 2009), and with semidefinite programming (SDP) in (Briales and Gonzalez-Jimenez, 2017). Absolute pose estimation was solved with the Gröbner basis method by several authors including Kneip et al. (2014), where the UPnP method was presented, and with semidefinite programming (SDP) (Agostinho et al., 2019).

Dynamical pose estimation was presented in (Yang et al., 2021a) as a unifying method for point cloud registration, primitive registration, category registration, and absolute pose estimation when there are no outliers. In this method the relative pose between the geometric primitives \mathbf{X}_i and \mathbf{Y}_i is found as the motion of a virtual rigid body that is initially aligned with the data set, and which is moved into alignment with the model set through the action of virtual springs and dampers connected between corresponding primitives \mathbf{X}_i and \mathbf{Y}_i .

In the case of outliers, RANSAC (Fischler and Bolles, 1981) has been widely used in registration problems. A drawback with RANSAC is slow convergence and low accuracy for high outlier fractions, as noted in (Parra Bustos and Chin, 2018), where a guaranteed outlier removal (GORE) technique was proposed. An alternative is to eliminate the need for RANSAC by replacing the least-squares cost $r(\cdot)^2$ with some robust cost $\rho(r(\cdot))$, like the Huber cost, the Geman-McClure cost or truncated least-squares cost (TLS) (Black and Rangarajan, 1996), which gives the minimization problem

$$\min_{\mathbf{T} \in SE(3)} \sum_{i=1}^N \rho(r(\mathbf{T} \otimes \mathbf{X}_i, \mathbf{Y}_i)) \quad (2)$$

The non-convexity of this problem was handled with semidefinite relaxation for the TLS case in the TEASER method (Yang et al., 2021b). Graduated non-convexity (GNC) (Blake and Zisserman, 1987), (Black and Rangarajan, 1996) is a continuation method for non-convex problems which was used with the Geman-McClure cost in (Zhou et al., 2016), and with the truncated least-squares cost (GNC-TLS) in (Yang et al., 2020). This method worked well for outlier rates up to 70%–90%.

Related Work

The GNC-TLS approach requires that a solver is available for the minimization problem (6) (Yang et al., 2020). Specialized solvers are available for the different types of problems, like point cloud registration where a closed form solution based on SVD is used (Arun et al., 1987), (Horn, 1987), primitive registration based on SDP (Briales and Gonzalez-Jimenez, 2017), absolute pose estimation with Gröbner basis methods (Kneip et al., 2014), and shape alignment with convex relaxation (Zhou et al., 2017) or a certifiable optimal solution based on SOS relaxation (Yang et al., 2020).

Contributions

In this paper, we present a new method where we combine the dynamical pose estimation method of Yang et al. (2021a) with the graduated non-convexity for truncated least-squares (GNC-TLS) as presented in (Yang et al., 2020). The main advantage of the proposed solution is that it is straightforward to implement, and that it gives a unified approach to 3D registration with outliers, including point cloud registration, primitive registration, and absolute pose estimation. Our solution is based on a systematic procedure for selecting the masses, spring constants and damping coefficients in the dynamical pose estimation scheme for each iteration of the graduated non-

convexity method. Moreover, Lyapunov analysis is included for point cloud registration to show that the method is stable and converges for each iteration. We demonstrate the performance of the method for point cloud registration, primitive registration and absolute pose estimation where the method is shown to perform well with up to 70%–90% outliers.

2 Graduated non-convexity

Consider the truncated least-squares (TLS) cost

$$\rho_{\text{TLS}}(r_i) = \min(r_i^2, \epsilon^2) \quad (3)$$

where ϵ is the truncation threshold. This cost gives a non-convex minimization problem (2). The graduated non-convexity (GNC) can be used to optimize this TLS problem in a GNC-TLS continuation process (Black and Rangarajan, 1996), (Antonante et al., 2022), (Yang et al., 2020), by minimization of the surrogate cost function

$$\rho_\mu(r_i) = w_i r_i^2 + \frac{\mu(1-w_i)}{\mu+w_i} \epsilon^2 \quad (4)$$

where the weights are given by

$$w_i = \begin{cases} 1, & \text{if } r_i^2 \leq \epsilon^2 \alpha_1 \\ \frac{\epsilon}{|r_i|} \sqrt{\mu(\mu+1)} - \mu, & \text{if } \epsilon^2 \alpha_1 \leq r_i^2 \leq \epsilon^2 \alpha_2 \\ 0, & \text{if } r_i^2 \geq \epsilon^2 \alpha_2 \end{cases} \quad (5)$$

for $\alpha_1 = \mu/(\mu+1)$ and $\alpha_2 = (\mu+1)/\mu$. Inserting the expressions for the weights (5) in (4) gives

$$\rho_\mu(r_i) = \begin{cases} r_i^2, & \text{if } r_i^2 \leq \epsilon^2 \alpha_1 \\ 2\epsilon|r_i|\sqrt{\mu(\mu+1)} - \mu(\epsilon^2 + r_i^2), & \text{if } \epsilon^2 \alpha_1 \leq r_i^2 \leq \epsilon^2 \alpha_2 \\ \epsilon^2, & \text{if } \epsilon^2 \alpha_2 \leq r_i^2 \end{cases}$$

It is seen that $\rho_\mu(r_i)$ equals $\rho_{\text{TLS}}(r)$ for $r_i^2 \leq \epsilon^2 \alpha_1$ and $\epsilon^2 \alpha_2 \leq r_i^2$, while there is a blend region for $\epsilon^2 \alpha_1 \leq r_i^2 \leq \epsilon^2 \alpha_2$ as shown in Figure 1. It is seen that $\rho_\mu''(r) = 2$ for small r , and $\rho_\mu''(r) = -2\mu$ in the blend region. This means that $\rho_\mu(r)$ will tend to a convex function when μ tends to zero.

It is noted in (Yang et al., 2020) that the second term in the cost function in (4) is independent from \mathbf{T} , and it follows that the GNC-TLS minimization problem given by (2) and (4) is equivalent to the weighted least-squares problem

$$\min_{\mathbf{T} \in SE(3)} \sum_{i=1}^N w_i r(\mathbf{T} \otimes \mathbf{X}_i, \mathbf{Y}_i)^2 \quad (6)$$

where the weights are given by (5).

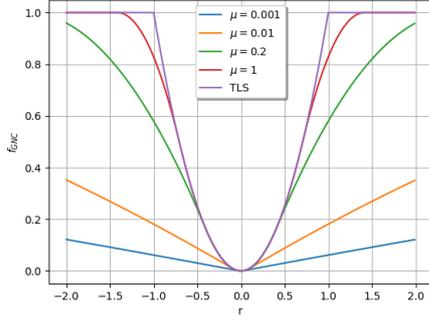


Figure 1: The surrogate cost function $\rho_\mu(r)$ for GNC-TLS for different values of the control parameter μ .

The TLS optimization problem (2) and (3) can then be solved in a GNC continuation process (Black and Rangarajan, 1996), (Yang et al., 2020) where the minimization problem (6) is solved first for a control variable μ that is sufficiently small for the problem to be approximately convex. For GNC-TLS μ can be initialized using

$$\mu = \frac{\epsilon^2}{2r_{\max}^2 - \epsilon^2} \quad (7)$$

where $r_{\max}^2 = \max_i r(\mathbf{T} \otimes X_i, Y_i)^2$ is the maximum residual after the first variable update (Yang et al., 2020). Then a sequence of minimization problems (6) is solved for increasing values of μ . The final solution is then found when the surrogate cost (4) is sufficiently close to the truncated least-squares cost (3). When μ becomes sufficiently large in this process the outliers will ideally have weight $w_i = 0$ and the inliers will ideally have $w_i = 1$, which means that only inliers will contribute to the quadratic cost in (6). The method is described in detail in (Antonante et al., 2022).

It is noted that the weighted least squares problem (6) can also be used for to solve GNC for the Geman-McClure cost (GNC-GM) (Zhou et al., 2016) by using the weights (Yang et al., 2020)

$$w_i = \left(\frac{\mu\epsilon^2}{r_i^2 + \mu\epsilon^2} \right)^2 \quad (8)$$

It is concluded that GNC-GM and GNC-TLS can be solved by minimizing (6). This requires that a solver is available for (6), and in this paper we will show how dynamical pose estimation (Yang et al., 2021a) can be modified so that it can be used as a solver.

3 Dynamical pose estimation for GNC-TLS

3.1 Introduction

Dynamical pose estimation (Yang et al., 2021a) is a solver for the minimization problem (1) based on a dynamic formulation where the primitives \mathbf{X}_i in the data set are modeled as point masses that are connected with virtual springs and dampers to the primitives \mathbf{Y}_i in the model set. The virtual springs will then pull the data set so that it aligns with the model set, and the displacement \mathbf{T} is found from the motion involved in the alignment. In the following we will present an extension of the original method of Yang et al. (2021a) which can be used as a solver for (6). This is done by letting the mass points have different masses determined by the weight w_i , and by scaling the spring and damper coefficients with the weights.

3.2 Dynamic model

The model primitives \mathbf{Y}_i are considered to be fixed in a rigid body \mathcal{B}_y and the data primitives \mathbf{X}_i are considered to be fixed in a rigid body \mathcal{B}_x , where \mathcal{B}_x is obtained by the rigid motion $\mathbf{T} \in SE(3)$ of \mathcal{B}_y . The rigid bodies \mathcal{B}_x and \mathcal{B}_y are fixed in the world frame s . Each primitive \mathbf{X}_i has a point mass of mass m_i at the position \mathbf{x}_i in the coordinates of s .

The position of mass point i in \mathcal{B}_x is given by

$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{x}_{r_i} \quad (9)$$

where $\bar{\mathbf{x}}$ is the center of mass of the rigid body \mathcal{B}_x . It is noted that

$$\bar{\mathbf{x}} = \frac{1}{M} \sum_{i=1}^N m_i \mathbf{x}_i, \quad \frac{1}{M} \sum_{i=1}^N m_i \mathbf{x}_{r_i} = \mathbf{0} \quad (10)$$

where $M = \sum_{i=1}^N m_i$ is the total mass. The moment of inertia of \mathcal{B}_x about its center of mass is

$$\mathbf{J} = - \sum_{i=1}^N m_i (\mathbf{x}_{r_i})^\times (\mathbf{x}_{r_i})^\times \quad (11)$$

where \mathbf{a}^\times is the skew-symmetric form of $\mathbf{a} \in \mathbb{R}^3$.

A moving virtual rigid body \mathcal{B}_z is defined as a copy of the rigid body \mathcal{B}_x . The body frame b is fixed in \mathcal{B}_z . The mass points at \mathbf{x}_i in \mathcal{B}_x are copied as the mass points at ζ_i in \mathcal{B}_z so that $\mathbf{x}_i = \zeta_i$, where \mathbf{x}_i is given in s and ζ_i is given in b . The virtual rigid body \mathcal{B}_z has center of mass with constant position $\bar{\zeta} = \bar{\mathbf{x}}$ in the body frame b . This gives $\zeta_i = \bar{\zeta} + \mathbf{x}_{r_i}$ where ζ_i and $\bar{\zeta}$ are given in the coordinates of b . It follows that the

moment of inertia of \mathcal{B}_z about its center of mass in the coordinates of b is \mathbf{J} as given by (11).

The displacement of frame b relative to frame s is given by $(\mathbf{R}_z(t), \mathbf{t}_z(t)) \in SE(3)$. Then the position of point i in \mathcal{B}_z in the s frame is

$$\mathbf{z}_i = \mathbf{R}_z \mathbf{x}_i + \mathbf{t}_z \quad (12)$$

The position of the center of mass of the rigid body \mathcal{B}_z in the s frame is therefore

$$\mathbf{z}_c = \frac{1}{M} \sum_{i=1}^N m_i \mathbf{z}_i \quad (13)$$

Then from (9), (10) and (12) it follows that

$$\mathbf{z}_i = \mathbf{z}_c + \mathbf{R}_z \mathbf{x}_{r_i}, \quad \dot{\mathbf{z}}_c = \mathbf{R}_z \dot{\mathbf{x}} + \dot{\mathbf{t}}_z \quad (14)$$

The virtual rigid body has velocity $\dot{\mathbf{z}}_c = \mathbf{v}_c$ for the center of mass, and angular velocity $\boldsymbol{\omega}$ in the coordinates of b . The initial values are $\mathbf{z}_c(0) = \bar{\mathbf{x}}$, $\mathbf{v}_c(0) = \mathbf{0}$, $\boldsymbol{\omega}(0) = \mathbf{0}$ and $\mathbf{R}_z(0) = \mathbf{I}$. For each of the N primitives there is one external force \mathbf{f}_i which acts on position \mathbf{z}_i . The equations of motion of the virtual rigid body \mathcal{B}_z are given by (Yang et al., 2021a)

$$\dot{\mathbf{z}}_c = \mathbf{v}_c \quad (15)$$

$$\dot{\mathbf{R}}_z = \mathbf{R}_z \boldsymbol{\omega}^\times \quad (16)$$

$$\dot{\mathbf{v}}_c = \mathbf{a}_c = \frac{1}{M} \mathbf{f} \quad (17)$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\alpha} = \mathbf{J}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega}^\times \mathbf{J} \boldsymbol{\omega}) \quad (18)$$

where \mathbf{J} , $\boldsymbol{\omega}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\tau}$ are given in b , and \mathbf{z}_c , \mathbf{v}_c and \mathbf{f} are given in s . The force \mathbf{f} on \mathcal{B}_z is given by $\mathbf{f} = \sum_{i=1}^N \mathbf{f}_i$ where \mathbf{f}_i is the force on particle i , and the torque is $\boldsymbol{\tau} = \sum_{i=1}^N \mathbf{x}_{r_i} \times (\mathbf{R}_z^T \mathbf{f}_i)$. Figure 2 in (Yang et al., 2021a) shows how the dynamics develop on a N -primitive rigid-body with $N = 4$ geometric primitives.

3.3 Graduated non-convexity with dynamical pose estimation

Dynamical pose estimation is formulated in (Yang et al., 2021a) as a minimization of the least-squares problem (1). Each primitive has equal mass $m_i = 1$, and all spring constants k_p and all damping coefficients k_d are equal. This means that $V_p = \frac{1}{2} k_p \sum_{i=1}^N \|\mathbf{r}(\mathbf{z}_i, \mathbf{y}_i)\|^2$ is minimized.

We present an extension where different masses m_i are used for the point masses, and where the spring and damper coefficients are scale with the mass using $k_{p,i} = m_i k_p$ and $k_{d,i} = m_i k_d$. Then by selecting $m_i = w_i$, the potential energy will be

$$V_p = \frac{1}{2} k_p \sum_{i=1}^N w_i \|\mathbf{r}(\mathbf{z}_i, \mathbf{y}_i)\|^2 \quad (19)$$

which means that the dynamical pose estimation method will minimize the GNC-TLS problem (6).

GNC-TLS can then be solved with DAMP as shown in Algorithm 1. The structure is similar to that of (Antonante et al., 2022), while the difference is the use of DAMP as the solver. The algorithm starts by initializing the control variable μ according to (7) and the weights w_i are all set to 1. For each iteration, dynamical pose estimation is used to solve (6). The weights are updated according to (5) and the control variable is increased. The algorithm terminates when either the maximum number of iterations are used, or when all inliers and outliers have been classified.

Algorithm 1: GNC-TLS

input : Model set $\mathbf{Y} = \{\mathbf{Y}_i\}_{i=1}^N$,
 Data set $\mathbf{X} = \{\mathbf{X}_i\}_{i=1}^N$,
 Initial pose $\mathbf{T}(0) = (\mathbf{R}_z(0), \mathbf{z}_c(0))$
 Truncation threshold $\epsilon > 0$,
 Control param. update factor $\gamma > 1$,
 Max iterations $j_{\max} > 0$

output: Final pose $\mathbf{T} = (\mathbf{R}_z, \mathbf{z}_c)$,
 Weights $\mathbf{w} = \{w_i\}_{i=1}^N$

- 1 $r_{\max}^2(0) = \max_i r(\mathbf{T}(0) \otimes \mathbf{X}_i, \mathbf{Y}_i)^2$
- 2 $\mu(0) = \text{initializeMu}(r_{\max}^2(0), \epsilon)$
- 3 $\mathbf{w}(0) = \{1\}_{i=1}^N$
- 4 **for** $j = 1$ **to** j_{\max} **do**
- 5 % DAMP solver shown in algorithm 2
- 6 $\mathbf{T} = \text{DAMP}(\mathbf{Y}, \mathbf{X}, \mathbf{T}, \mathbf{w})$
- 7 **if** \mathbf{w} is binary **and** $j > 1$ **then**
- 8 | break
- 9 **else**
- 10 | $r^2 = \{r(\mathbf{T} \otimes \mathbf{X}_i, \mathbf{Y}_i)^2\}_{i=1}^N$
- 11 | $\mathbf{w} = \text{updateWeights}(r^2, \epsilon, \mu)$
- 12 | $\mu = \gamma \cdot \mu$
- 13 **end**
- 14 **end**

The damp method used in Algorithm 1 was implemented with Algorithm 2, which is based on the algorithm of (Yang et al., 2021a). The algorithm starts by initializing the particle point mass information for the moving virtual body. An initial pose is passed to the algorithm giving faster convergence, as past iterations of GNC-TLS are utilized, the weights w_i converge, and the pose estimate approaches the final solution. In the main loop, the moving virtual rigid body \mathcal{B}_z is moved and the forces for each particle are calculated using the distance function defined for the given primitive correspondence pair. The total force and torques are calculated, and the states of the dynamic system are updated. The algorithm terminates when either the

maximum number of iterations are used, or the change in the states is below a set threshold. In (Yang et al., 2021a) the *EscapeMinimum* scheme is presented to prevent DAMP from returning a sub-optimal solution. If *EscapeMinimum* is used, the solution is recorded and its corresponding potential energy is saved. The states are then randomly perturbed, before a new dynamical simulation is made. This is repeated for a given number of trials, and the solution that is returned is the pose estimate that yields the lowest potential energy in the system. We have modified this scheme to only perturb the linear and angular velocities.

3.4 Point cloud registration

In point cloud registration the geometric primitives are points with position \mathbf{y}_i and \mathbf{x}_i , and the residual in (6) is $r(\mathbf{T} \otimes \mathbf{X}_i, \mathbf{Y}_i) = \|\mathbf{y}_i - \mathbf{R}\mathbf{x}_i - \mathbf{t}\|$. It is noted that in this case (6) is Wahba's problem with weights, which has a closed form solution which is a straightforward extension of the result in (Arun et al., 1987). To solve this problem with dynamical pose estimation, we make a slight modification of the approach in (Yang et al., 2021a) by allowing for different masses m_i , and select the force acting on the point mass at \mathbf{z}_i as

$$\mathbf{f}_i = m_i k_p (\mathbf{y}_i - \mathbf{z}_i) - m_i k_d \dot{\mathbf{z}}_i \quad (20)$$

This is the force of a spring with spring constant $m_i k_p$ and a damper with coefficient $m_i k_d$. The resulting torque about the center of mass of \mathcal{B}_z is

$$\begin{aligned} \boldsymbol{\tau}_i &= \mathbf{x}_{r_i} \times (\mathbf{R}_z^T \mathbf{f}_i) \\ &= m_i \mathbf{x}_{r_i} \times \mathbf{R}^T (k_p (\mathbf{y}_i - \mathbf{z}_i) - k_d \dot{\mathbf{z}}_i) \end{aligned} \quad (21)$$

Using (14), $\dot{\mathbf{z}}_i = \mathbf{v}_c + \mathbf{R}_z \boldsymbol{\omega} \times \mathbf{x}_{r_i}$ is derived. Combined with (10) applied to the point masses in \mathcal{B}_y , the total force $\mathbf{f} = \sum_{i=1}^N \mathbf{f}_i$ and the total torque $\boldsymbol{\tau} = \sum_{i=1}^N \boldsymbol{\tau}_i$ on the rigid body \mathcal{B}_z is found to be (Yang et al., 2021a)

$$\mathbf{f} = M k_p (\bar{\mathbf{y}} - \mathbf{z}_c) - M k_d \mathbf{v}_c \quad (22)$$

$$\boldsymbol{\tau} = k_p \sum_{i=1}^N m_i \mathbf{x}_{r_i} \times \mathbf{R}_z^T \mathbf{y}_{r_i} - k_d \mathbf{J} \boldsymbol{\omega} \quad (23)$$

We note that the residual used in point cloud registration is the spring deflection given by

$$\mathbf{r}(\mathbf{z}_i, \mathbf{y}_i) = \mathbf{y}_i - \mathbf{z}_i \quad (24)$$

and that the resulting potential energy is given by (19).

3.5 Convergence for point cloud registration

The stability and convergence of dynamic pose estimation for point cloud registration can be studied with

Algorithm 2: DAMP for GNC

```

input : Model set  $\mathbf{Y} = \{\mathbf{Y}_i\}_{i=1}^N$ , Data set
          $\mathbf{X} = \{\mathbf{X}_i\}_{i=1}^N$ , Initial pose
          $\mathbf{T}(0) = (\mathbf{R}_z(0), \mathbf{z}_c(0))$ , Weights (mass)
          $\mathbf{w} = \{w_i\}_{i=1}^N > 0$ , Spring and
         damping coeff.  $k_s > 0, k_d > 0$ , Time
         step  $\Delta t > 0$ , Max iter.  $k_{\max} > 0$ ,
         Stopping criteria  $\sigma > 0$ ,
         EscapeMinimum TRUE or FALSE,
         Max EscapeMinimum trials  $t_{\max} > 0$ 

output: Final pose  $\mathbf{T} = (\mathbf{R}_z, \mathbf{z}_c)$ ,

1  $M = \sum_{i=1}^N w_i, \bar{\mathbf{x}} = \frac{1}{M} \sum_{i=1}^N w_i \mathbf{x}_i$ 
2  $\mathbf{x}_{r_i} = \mathbf{x}_i - \bar{\mathbf{x}}, \mathbf{J} = - \sum_{i=1}^N w_i (\mathbf{x}_{r_i})^\times (\mathbf{x}_{r_i})^\times$ 
3  $\mathbf{v}_c(0) = \mathbf{0}, \boldsymbol{\omega}(0) = \mathbf{0}$ 
4 if EscapeMinimum then
5   |  $\text{trial} = 0, \mathcal{T} = \emptyset, \mathcal{V} = \emptyset$ 
6 end
7 for  $k = 1$  to  $k_{\max}$  do
8   |  $\mathbf{z}_i = \mathbf{R}_z \cdot \mathbf{x}_{r_i} + \mathbf{z}_c, i = 1, \dots, N$ 
9   |  $\mathbf{f}_i = w_i (k_s r(\mathbf{z}_i, \mathbf{y}_i) - k_d \dot{\mathbf{z}}_i), i = 1, \dots, N$ 
10  |  $\mathbf{f} = \sum_{i=1}^N \mathbf{f}_i$ 
11  |  $\boldsymbol{\tau} = \sum_{i=1}^N \mathbf{x}_{r_i} \times \mathbf{R}_z^T \mathbf{f}_i$ 
12  | % Update states ( $s = s + \Delta t \cdot \dot{s}$ )
13  |  $\mathbf{z}_c = \mathbf{z}_c + \Delta t \cdot \mathbf{v}_c$ 
14  |  $\mathbf{R}_z = \mathbf{R}_z \cdot \exp(\Delta t \cdot \boldsymbol{\omega})$ 
15  |  $\mathbf{v}_c = \mathbf{v}_c + \Delta t \cdot \frac{1}{M} \mathbf{f}$ 
16  |  $\boldsymbol{\omega} = \boldsymbol{\omega} + \Delta t \cdot \mathbf{J}^{-1} (\boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{J} \boldsymbol{\omega})$ 
17  | % Equilibrium point
18  | if  $\|\dot{s}\| \leq \sigma$  then
19    | if EscapeMinimum and  $\text{trial} \leq t_{\max}$ 
20      | then
21        |  $\mathcal{T} = \mathcal{T} \cup (\mathbf{R}_z, \mathbf{z}_c)$ 
22        |  $\mathcal{V} = \mathcal{V} \cup \frac{k_s}{2} \sum_{i=1}^N w_i \|r(\mathbf{z}_i, \mathbf{y}_i)\|^2$ 
23        |  $\{\mathbf{v}_c, \boldsymbol{\omega}\} \sim \mathcal{N}(0, I)$ 
24        |  $\text{trial} = \text{trial} + 1$ 
25      | else
26        | break
27      | end
28    | end
29  | if EscapeMinimum then
30    |  $(\mathbf{R}_z, \mathbf{z}_c) = \mathcal{T}(\arg \min \mathcal{V})$ 
31  | end

```

LaSalle's invariance principle (Khalil, 2002). The problem formulation leads to a virtual mass-spring-damper system which is well known in energy-based Lyapunov analysis. In (Yang et al., 2021a) the details of the Lyapunov analysis are left out, while a detailed analysis of the equilibrium points is included. In (Yang et al.,

2021a, Theorem 11) it is shown that one optimal equilibrium points is stable, while three other equilibrium points which differ from the optimal equilibrium point by a rotation of π , are unstable.

Consider the Lyapunov function candidate

$$V = V_k + V_p \quad (25)$$

where the kinetic energy V_k of the rigid body \mathcal{B}_z is (Egeland and Gravdahl, 2002)

$$V_k = \frac{1}{2} \sum_{i=1}^N m_i \dot{\mathbf{z}}_i^T \dot{\mathbf{z}}_i = \frac{1}{2} M \mathbf{v}_c^T \mathbf{v}_c + \frac{1}{2} \boldsymbol{\omega}^T \mathbf{J} \boldsymbol{\omega}$$

and the potential energy V_p stored in the springs is given by (19) for point cloud registration

$$V_p = \frac{1}{2} k_p \sum_{i=1}^N m_i \|\mathbf{y}_i - \mathbf{z}_i\|^2 \quad (26)$$

Note that the point masses m_i are included in the expression for the potential energy. This is explained in Section 3.3.

The time derivative of the kinetic energy along the trajectories of the system (17, 18) is

$$\dot{V}_k = \mathbf{v}_c^T M \dot{\mathbf{v}}_c + \boldsymbol{\omega}^T \mathbf{J} \dot{\boldsymbol{\omega}} = \mathbf{v}_c^T \mathbf{f} + \boldsymbol{\omega}^T \boldsymbol{\tau} \quad (27)$$

where it is used that $\boldsymbol{\omega}^T (\boldsymbol{\omega} \times \mathbf{J} \boldsymbol{\omega}) = \boldsymbol{\omega}^T \boldsymbol{\omega} \times (\mathbf{J} \boldsymbol{\omega}) = \mathbf{0}$. The time derivative along the solutions of the closed-loop system (17, 18, 22, 23) is therefore

$$\begin{aligned} \dot{V}_k &= k_p M \mathbf{v}_c^T (\bar{\mathbf{y}} - \mathbf{z}_c) - k_d M \mathbf{v}_c^T \mathbf{v}_c \\ &+ k_p \boldsymbol{\omega}^T \sum_{i=1}^N m_i \mathbf{x}_{r_i}^\times \mathbf{R}_z^T \mathbf{y}_{r_i} - k_d \boldsymbol{\omega}^T \mathbf{J} \boldsymbol{\omega} \end{aligned} \quad (28)$$

The expression for the potential energy of spring i is modified by using (14) and (9, 10) applied to both \mathcal{B}_x and \mathcal{B}_y , which gives

$$V_p = \frac{1}{2} k_p M \|\mathbf{y}_i - \mathbf{z}_i\|^2 + \frac{1}{2} \sum_{i=1}^N m_i \|\mathbf{y}_{r_i} - \mathbf{R}_z \mathbf{x}_{r_i}\|^2 \quad (29)$$

The time derivative of the potential energy along the solutions of the system (17, 18, 22, 23) is then

$$\dot{V}_p = -k_p M \mathbf{v}_c^T (\bar{\mathbf{y}} - \mathbf{z}_c) - k_p \boldsymbol{\omega}^T \sum_{i=1}^N m_i \mathbf{x}_{r_i}^\times \mathbf{R}_z^T \mathbf{y}_{r_i} \quad (30)$$

The time derivative of the Lyapunov function candidate along the solutions of the system (17, 18, 22, 23) is therefore

$$\dot{V} = -k_d M \mathbf{v}_c^T \mathbf{v}_c - k_d \boldsymbol{\omega}^T \mathbf{J} \boldsymbol{\omega} = -2k_d V_k \leq 0 \quad (31)$$

Stability can then be analyzed with LaSalle's invariance principle. From

$$\dot{\mathbf{v}}_c = \frac{\mathbf{f}}{M} = k_p (\bar{\mathbf{y}} - \mathbf{z}_c) - k_d \mathbf{v}_c \quad (32)$$

it is seen that invariance at $\mathbf{v}_c = \mathbf{0}$ requires that $\bar{\mathbf{y}} = \mathbf{z}_c$. This ensures that the center of mass of \mathcal{B}_z converges to the center of mass of \mathcal{B}_y . From

$$\mathbf{J} \dot{\boldsymbol{\omega}} = k_p \sum_{i=1}^N m_i \mathbf{x}_{r_i}^\times \mathbf{R}_z^T \mathbf{y}_{r_i} - k_d \mathbf{J} \boldsymbol{\omega} - \boldsymbol{\omega} \times \mathbf{J} \boldsymbol{\omega} \quad (33)$$

it is seen that invariance at $\boldsymbol{\omega} = \mathbf{0}$ requires that

$$\sum_{i=1}^N m_i \mathbf{x}_{r_i}^\times \mathbf{R}_z^T \mathbf{y}_{r_i} = \sum_{i=1}^N m_i \mathbf{x}_{r_i}^\times \mathbf{R}_z^T \mathbf{R} \mathbf{x}_{r_i} = \mathbf{0} \quad (34)$$

This is the case for $\mathbf{R} = \mathbf{R}_z$, but also for $\mathbf{R} = \mathbf{R}_z \mathbf{R}(\mathbf{k}, \pi)$, where $\mathbf{R}(\mathbf{k}, \pi)$ is the rotation matrix of a rotation π about some axis \mathbf{k} . In (Yang et al., 2021a) it is shown with local stability theory that solutions with $\mathbf{R} = \mathbf{R}_z \mathbf{R}(\mathbf{k}, \pi)$ are unstable. This means that the points \mathbf{z}_i will converge to the points \mathbf{y}_i , which means that the solution is found from the final value of \mathbf{R}_z and \mathbf{z}_c .

3.6 Primitive registration

Primitive registration with dynamical pose estimation was proposed in (Yang et al., 2021a), where \mathbf{Y}_i could be points, lines or planes.

The first type of primitive is a point $\mathbf{y}_i \in \mathbb{R}^3$. The residual is then as in point cloud registration.

The second primitive is a line

$$L(\mathbf{y}_i, \mathbf{a}_i) = \{\mathbf{y}_i + \alpha_i \mathbf{a}_i : \alpha_i \in \mathbb{R}\} \quad (35)$$

where $\mathbf{y}_i \in \mathbb{R}^3$ is a point on the line and $\mathbf{a}_i \in \mathbb{S}^2$ is a unit direction vector. The spring deflection is then the distance from a point \mathbf{z}_i to the corresponding line $L(\mathbf{y}_i, \mathbf{a}_i)$, which is

$$\mathbf{r}(\mathbf{z}_i, \mathbf{y}_i) = (\mathbf{I} - \mathbf{a}_i \mathbf{a}_i^T) (\mathbf{z}_i - \mathbf{y}_i) \quad (36)$$

The third primitive is a plane

$$H(\mathbf{y}_i, \mathbf{n}_i) = \{\mathbf{y} \in \mathbb{R}^3 : (\mathbf{y} - \mathbf{y}_i) \cdot \mathbf{n}_i = 0\} \quad (37)$$

where \mathbf{y}_i and \mathbf{y} are points in the plane and $\mathbf{n}_i \in \mathbb{S}^2$ is the unit normal vector. The spring deflection is then the distance from the point \mathbf{z}_i to the corresponding plane $H(\mathbf{y}_i, \mathbf{n}_i)$, which is

$$\mathbf{r}(\mathbf{z}_i, \mathbf{y}_i) = \mathbf{n}_i \mathbf{n}_i^T (\mathbf{z}_i - \mathbf{y}_i) \quad (38)$$

The force on particle i is for all 3 cases

$$\mathbf{f}_i = -m_i k_p \mathbf{r}(\mathbf{z}_i, \mathbf{y}_i) - m_i k_d \dot{\mathbf{z}}_i \quad (39)$$

and the resulting potential energy is given by (19) where $\mathbf{r}(\mathbf{z}_i, \mathbf{y}_i)$ is given by (24) for a point-point correspondence, (36) for a point-line correspondence, and by (38) for a point-plane correspondence.

3.7 Absolute pose estimation

In absolute pose estimation the model points \mathbf{y}_i and the corresponding data points \mathbf{x}_i are given in the camera frame. The primitives of the model are the points \mathbf{y}_i , while the primitives of the data set are the lines $L(\mathbf{0}, \mathbf{a}_i)$ through the origin of the camera frame with direction vector $\mathbf{a}_i = \mathbf{p}_i / \|\mathbf{p}_i\|$, where \mathbf{p}_i are the image points corresponding to the points \mathbf{x}_i (Hartley and Zisserman, 2004). The virtual rigid body \mathcal{B}_z is initially aligned with \mathcal{B}_y , and is moved to \mathcal{B}_x by connecting each point \mathbf{z}_i to the closest point on the corresponding line $L(\mathbf{0}, \mathbf{a}_i)$ with a spring. Since the lines will spread out from the origin of frame c , this alignment will also give the right depth coordinates of the data points.

The residual is given by (36) as in the point-line case, which gives $\mathbf{r}(\mathbf{z}_i, \mathbf{y}_i) = (\mathbf{I} - \mathbf{a}_i \mathbf{a}_i^T) \mathbf{z}_i$. The force on particle i is then

$$\mathbf{f}_i = -m_i k_p (\mathbf{I} - \mathbf{a}_i \mathbf{a}_i^T) \mathbf{z}_i - m_i k_d \dot{\mathbf{z}}_i \quad (40)$$

The resulting potential energy is given by (19).

4 Experiments

To validate the performance of the proposed method, three different registration problems were simulated. The results are presented in the following section. The method was implemented in MATLAB using for-loops, cell arrays and structure array to handle the different primitive types. The implementation is without third party libraries. This impacts the computation time of the method, and a significant reduction in computation time is expected with an optimized implementation in C++. The experiments were run for a range of outlier fractions, starting at 0% outliers, and incrementally increasing up to 90% outliers. For each outlier fraction 20 Monte Carlo runs were simulated. The experiments were run on a system with an Intel Core i7-8700 CPU.

The GNC-TLS was solved with dynamical pose estimation by running a sequence of iterations $j = 0, \dots, n_{it}$. The weights were given by $w_i = 1$ for $j = 0$, and by (5) for $j > 0$. The initial value for the control parameter μ_j was set using (7). The control parameter was increased for each iteration according to $\mu_{j+1} = \gamma \mu_j$, where $\gamma = 1.4$ as in (Yang et al., 2020). The spring constant was set to $k_p = 2$, as done by (Yang et al., 2021a), and the damper constant was set to $k_d = 2\omega_0$, where $\omega_0 = \sqrt{k_p}$, for a critically damped system.

4.1 Point cloud registration

In point cloud registration the point-to-point correspondence is used, as both the model set $\{Y_i\}_{i=1}^N$ and the data set $\{X_i\}_{i=1}^N$ consist of corresponding 3D points. The performance of Dynamical Pose Estimation with Graduated Non-Convexity in point cloud registration was investigated. An experiment using the *Stanford Bunny* from the The Stanford 3D Scanning Repository (Curless and Levoy, 1996) was conducted following the procedure in (Yang et al., 2020) (Figure 2). The Bunny point cloud was scaled to fit inside a unit cube to create the model set $\{Y_i\}_{i=1}^N$. A random rigid transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in SE(3)$ restricted to the scene radius $\|\mathbf{t}\| = 1$ was applied to the model. Zero mean Gaussian noise with the standard deviation $\sigma = 0.01$ was added to create the data set $\{X_i\}_{i=1}^N$. A subset of the given correspondences between points in the model set and data set is chosen at random, limited to $N = 100$ correspondences. As in (Yang et al., 2020) and (Yang and Carlone, 2019), a given fraction of the model set points are replaced by points uniformly sampled from a sphere with radius $r = 2$ to create outliers for the problem. The DAMP stopping criterion is set to $\|\dot{\mathbf{s}}\| \leq 1e^{-4}$ and *EscapeMinimum* (Yang et al., 2021a) is not used. As in (Antonante et al., 2022), the truncation threshold ϵ for TLS-GNC is set to $\epsilon = \sigma \sqrt{\chi^2_{inv}(0.99, 3)} = 0.0337$.

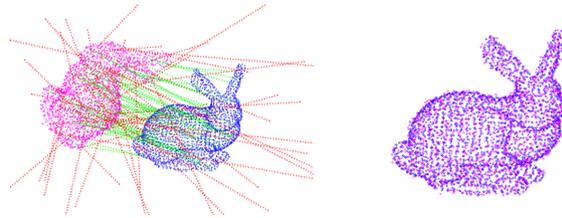


Figure 2: Point cloud registration for the Stanford Bunny. Green and red lines represent inlier and outlier correspondences, respectively

The results of the experiment are presented in Figure 4. The experiment showed that Dynamical Pose Estimation with Graduated Non-Convexity is robust with up to 90% outliers. Compared to the results obtained in (Yang et al., 2021a), the order of magnitude of both the rotation errors and translation errors are comparable, even though we have up to 90% outliers. This result also outperforms what is reported in (Yang et al., 2020) for GNC-TLS. The number of iterations the method uses to converge is similar to (Yang et al., 2020), and the method takes 0.101 s to run on the given computer system with the MATLAB implementation.

4.2 Primitive registration

In primitive registration the correspondences are between different geometric primitives in the model set $\{Y_i\}_{i=1}^N$ and points of the measured data set $\{X_i\}_{i=1}^N$ are used (Yang et al., 2021a). The performance of Dynamical Pose Estimation with Graduated Non-Convexity in primitive registration was evaluated in an experiment with point-to-point, point-to-line, and point-to-plane correspondences. The experiment followed the setup in (Antonante et al., 2022). The *aeroplane-2* mesh model from the PASCAL3D+ dataset (Xiang et al., 2014) was used (Figure 3). The model set was assembled from the mesh model as a set of 58 499 vertices, 126 813 edges, and 68 993 faces. The point cloud of the data set was created by sampling points on the vertices, edges and faces of the mesh model. A rigid transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in SE(3)$ restricted to the scene radius $\|\mathbf{t}\| = 1$ was applied to the sampled points, and zero mean Gaussian noise with the standard deviation $\sigma = 0.005 d_{\text{mesh}}$ was added. The diameter of the aeroplane-2 model is $d_{\text{mesh}} \approx 0.76$, resulting in noise with standard deviation $\sigma = 0.0038$. Correspondences between the model and data sets were set up, randomly selecting 40 point-to-point, 80 point-to-line, and 80 point-to-plane correspondences between the model and the data, creating $N = 200$ correspondences. Outliers were added by imposing false correspondences between a given fraction of the 200 selected points and randomly selected primitives in the model set that were not a part of original 200 correspondences. The DAMP stopping criterion was set to $\|\dot{\mathbf{s}}\| \leq 1e^{-4}$ and *EscapeMinimum* (Yang et al., 2021a) was used, allowing a re-run of DAMP one additional time following a random perturbation of the linear and angular velocities of the rigid body system. As in (Antonante et al., 2022), the truncation threshold ϵ for TLS-GNC was given by $\epsilon = \sigma \sqrt{\text{chi2inv}(0.99, 3)} = 0.0128$.

The results from the experiment are presented in Figure 5. The results show that Dynamical Pose Estimation with Graduated Non-Convexity was robust up to 80% outliers when applied to primitive registration. The errors are comparable to those shown in (Yang et al., 2020), as are the number of iterations used to return a solution. Our method takes on average 0.739 s to converge on the stated computer system with the unoptimized MATLAB implementation. The use of *EscapeMinimum* in DAMP is needed as some unsuccessful runs were observed without it.

4.3 Absolute pose estimation

In absolute pose estimation a set of model points are aligned with a set of bearing vectors as defined by

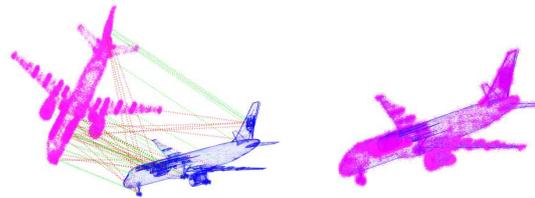


Figure 3: Primitive registration problem for the Aeroplane-2 mesh model. Green and red lines represent inlier and outlier correspondences respectively

image points and the camera frame. For this problem the point-to-line correspondence is used. The performance of Dynamical Pose Estimation with Graduated Non-Convexity in absolute pose estimation was evaluated in an experiment following the setup used in (Yang et al., 2021a) with the inclusion of outliers. $N = 200$ points in 3D were uniformly sampled from a $[-2, 2] \times [-2, 2] \times [4, 8]$ box in the camera frame. The points were projected into the image plane and zero mean Gaussian noise with standard deviation $\sigma = 0.01$ was added to the projections. The bearing vectors were formed from the noisy projections, creating the data set $\{X_i\}_{i=1}^N$. The original point cloud was transformed with a random rigid transformation $\mathbf{T} = (\mathbf{R}, \mathbf{t}) \in SE(3)$ restricted to the scene radius $\|\mathbf{t}\| = 1$ to create the model set $\{Y_i\}_{i=1}^N$. Outliers were added to the data set according to the given outlier fraction, by creating new bearing vectors from new 3D points sampled in the same $[-2, 2] \times [-2, 2] \times [4, 8]$ box in the camera frame. The DAMP stopping criterion was set to $\|\dot{\mathbf{s}}\| \leq 1e^{-4}$ and *EscapeMinimum* (Yang et al., 2021a) was used, allowing a re-run of DAMP three additional times following a random perturbation of the linear and angular velocities of the rigid body system each time. The GNC-TLS truncation threshold was set to $\epsilon = \sigma \sqrt{\text{chi2inv}(0.99, 3)} = 0.0337$.

The results from the experiment are presented in Figure 6. We observe that Dynamical Pose Estimation with Graduated Non-Convexity is able to return an accurate solution for up to 70% outliers, but fails when the outlier fraction increases beyond 70%. There is a major computational cost to running DAMP with *EscapeMinimum* enabled, but it is required for robustness of the algorithm, as the absolute pose estimation problem is reported in (Yang et al., 2021a) to be a challenging problem for DAMP. The algorithm uses on average 5.368 s to return a solution in the MATLAB implementation.

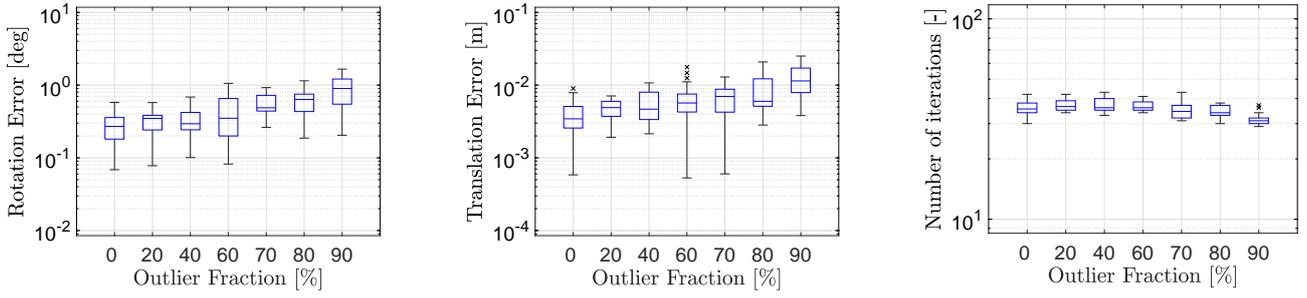


Figure 4: Box plots showing rotation errors, translation errors, and number of iterations used for Point cloud registration using the Stanford Bunny data set, solved using Dynamical Pose Estimation with Graduated Non-Convexity.

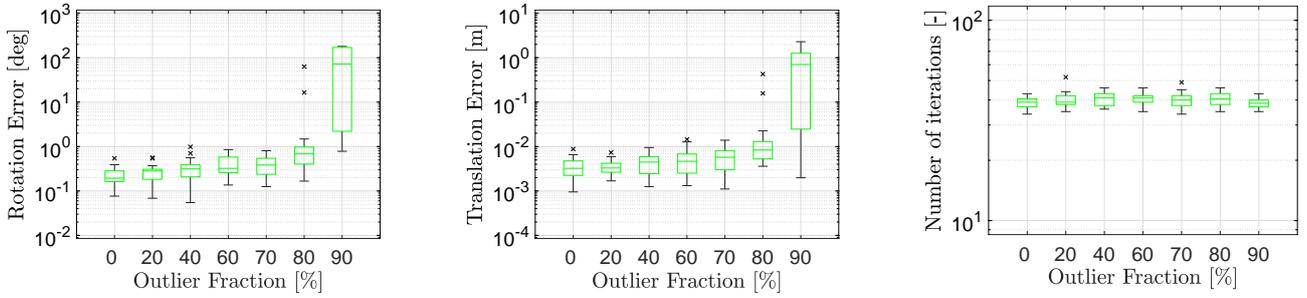


Figure 5: Box plots showing rotation errors, translation errors, and number of iterations used for Primitive registration using the Aeroplane-2 mesh model in the PASCAL3D+ data set, solved using Dynamical Pose Estimation with Graduated Non-Convexity.

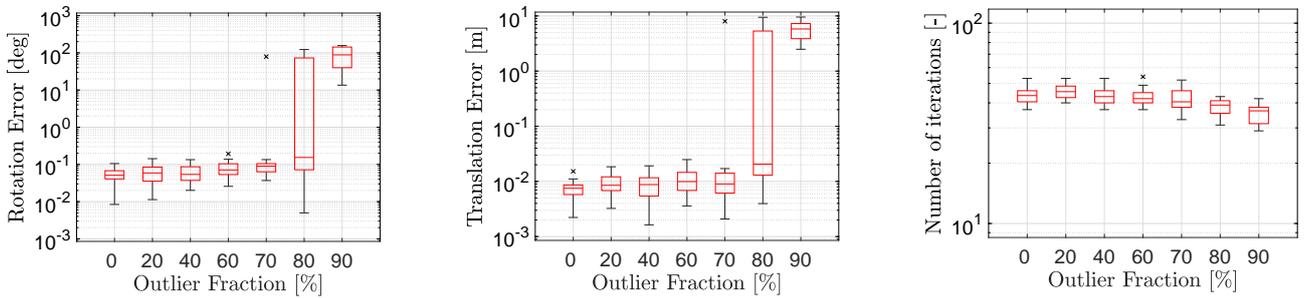


Figure 6: Box plots showing rotation errors, translation errors, and number of iterations used for Absolute pose estimation using a synthetic sample problem, solved using Dynamical Pose Estimation with Graduated Non-Convexity.

Acknowledgments

We would like to thank Dr. Heng Yang for providing the code used in his implementations.

This work was funded by the Norwegian Research Council under Project Number 237896, SFI Offshore Mechatronics.

References

- Agostinho, S., Gomes, J., and Del Bue, A. Cvx-PnP: A unified convex solution to the absolute pose estimation problem from point and line correspondences, 2019. ArXiv preprint arXiv:1907.10545 [cs.CV].
- Antonante, P., Tzoumas, V., Yang, H., and Carlone, L. Outlier-robust estimation: Hardness, minimally tuned algorithms, and applications. *IEEE Transactions on Robotics (T-RO)*, 2022. 38(1):281–301. doi:[10.1109/TRO.2021.3094984](https://doi.org/10.1109/TRO.2021.3094984).
- Arun, K., Huang, T., and Blostein, S. Least-squares filtering of two 3-D point sets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1987. PAMI-9(5):698 – 700. doi:[10.1109/TPAMI.1987.4767965](https://doi.org/10.1109/TPAMI.1987.4767965).
- Black, M. J. and Rangarajan, A. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 1996. 19(1):57–91. doi:[10.1007/BF00131148](https://doi.org/10.1007/BF00131148).
- Blake, A. and Zisserman, A. *Visual Reconstruction*. The MIT Press, 1987. doi:[10.7551/mitpress/7132.001.0001](https://doi.org/10.7551/mitpress/7132.001.0001).
- Briales, J. and Gonzalez-Jimenez, J. Convex global 3D registration with Lagrangian duality. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pages 5612–5621, 2017. doi:[10.1109/CVPR.2017.595](https://doi.org/10.1109/CVPR.2017.595).
- Curless, B. and Levoy, M. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96*. Association for Computing Machinery, New York, NY, USA, page 303–312, 1996. doi:[10.1145/237170.237269](https://doi.org/10.1145/237170.237269).
- Egeland, O. and Gravdahl, J. T. *Modeling and Simulation for Automatic Control*. Marine Cybernetics, 2002.
- Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 24(6):381–395. doi:[10.1145/358669.358692](https://doi.org/10.1145/358669.358692).
- Hartley, R. and Kahl, F. Global optimization through rotation space search. *International Journal of Computer Vision*, 2009. 82(1):64–79. doi:[10.1007/s11263-008-0186-9](https://doi.org/10.1007/s11263-008-0186-9).
- Hartley, R. I. and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. doi:[10.1017/CBO9780511811685](https://doi.org/10.1017/CBO9780511811685).
- Horn, B. K. P. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 1987. 4(4):629–642. doi:[10.1364/JOSAA.4.000629](https://doi.org/10.1364/JOSAA.4.000629).
- Khalil, H. K. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002.
- Kneip, L., Li, H., and Seo, Y. UPnP: An optimal $O(n)$ solution to the absolute pose problem with universal applicability. In *Proceedings of the European Conf. on Computer Vision (ECCV)*. Springer, pages 127–142, 2014. doi:[10.1007/978-3-319-10590-1_9](https://doi.org/10.1007/978-3-319-10590-1_9).
- Olsson, C., Kahl, F., and Oskarsson, M. Branch-and-bound methods for Euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009. 31(5):783–794. doi:[10.1109/TPAMI.2008.131](https://doi.org/10.1109/TPAMI.2008.131).
- Parra Bustos, Á. and Chin, T. J. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 40(512):2868–2882. doi:[10.1109/TPAMI.2017.2773482](https://doi.org/10.1109/TPAMI.2017.2773482).
- Xiang, Y., Mottaghi, R., and Savarese, S. Beyond PASCAL: A benchmark for 3D object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pages 75–82, 2014. doi:[10.1109/WACV.2014.6836101](https://doi.org/10.1109/WACV.2014.6836101).
- Yang, H., Antonante, P., Tzoumas, V., and Carlone, L. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters (RA-L)*, 2020. 5(2):1127–1134. doi:[10.1109/LRA.2020.2965893](https://doi.org/10.1109/LRA.2020.2965893).
- Yang, H. and Carlone, L. A polynomial-time solution for robust registration with extreme outlier

- rates. In *Proceedings of Robotics: Science and Systems (RSS)*. Freiburg im Breisgau, Germany, 2019. doi:[10.15607/RSS.2019.XV.003](https://doi.org/10.15607/RSS.2019.XV.003).
- Yang, H., Doran, C., and Slotine, J.-J. Dynamical pose estimation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. pages 5906–5915, 2021a. doi:[10.1109/ICCV48922.2021.00587](https://doi.org/10.1109/ICCV48922.2021.00587).
- Yang, H., Shi, J., and Carlone, L. TEASER: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 2021b. 37(2):314 – 333. doi:[10.1109/TRO.2020.3033695](https://doi.org/10.1109/TRO.2020.3033695).
- Zhou, Q., Park, J., and Koltun, V. Fast global registration. In *European Conference on Computer Vision (ECCV)*. Springer, pages 766–782, 2016. doi:[10.1007/978-3-319-46475-6_47](https://doi.org/10.1007/978-3-319-46475-6_47).
- Zhou, X., Zhu, M., Leonardos, S., and Daniilidis, K. Sparse representation for 3D shape estimation: A convex relaxation approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 39(8):1648–1661. doi:[10.1109/TPAMI.2016.2605097](https://doi.org/10.1109/TPAMI.2016.2605097).