SoftPOSIT Enhancements for Monocular Camera Spacecraft Pose Estimation

Jian-Feng Shi
Carleton University
Ottawa, Ontario K1S 5B6, Canada
Email: jianfeng.shi@carleton.ca

Steve Ulrich
Carleton University
Ottawa, Ontario K1S 5B6, Canada
Email: steve.ulrich@carleton.ca

***Abstract*—This paper proposes several enhancements to the *softPOSIT* algorithm with applications to spacecraft pose estimation using a monocular camera. First, the proposed enhancements include a technique for reducing false matches as result of local minimum trapping. Second, this paper provides two strategies for iteration control parameter initialization by using the trace of the correspondence distance, and by using image centroid matching. The method of image centroid matching allows the world model center of geometry to align with the image centroid. The alignment result in reasonable correspondence weighting values used for match optimization. The various algorithm enhancements were tested on $26,180$ simulations with varying geometries and initial pose conditions. Results show a significant increase in accuracy when compared with the original method.**

I. Introduction

The technology of using camera images for robotic and unmanned aerial vehicle guidance, navigation, and control (GNC) have significantly matured in the recent years. For example, the advances in vision techniques can benefit multi-spacecraft formation flying missions [1] using monocular optical and thermal cameras as proximity operation navigation sensors. Pose estimation using monocular camera images determines position and orientation of the target spacecraft relative to the observer camera coordinate system. Many techniques are available in solving the pose estimation problem although the most advanced techniques based on complex non-geometric features requires significant computational resources and are less efficient for some space applications where the target image can be approximated by simple geometries [2]. A more suitable approach is by finding correspondence between the internal 3D model with the 2D camera image hence inferring the target pose. This is called the *model-to-image registration problem* or *simultaneous pose and correspondence problem*. The image point correspondence is also known as the *Perspective-n-Point* (PnP) problem. [3] For the unknown space vehicle, the focus remains to solve the arbitrary n point problem. Some selected solution to the PnP problem include the use of numerical scheme solutions such as RANdom SAmple Consensus (RANSAC) [4], Iterative Closest Point (ICP) [5], Newton-Raphson Method (NRM) [6], and non-iterative solution such as the EPnP [7] solution that is on the order of $O(n)$.

David *et al*. [8] solve the PnP problem by combining Simulated Annealing (SA) and Scaled Orthographic Projection (SOP) [9]. This method is termed *SoftPOSIT* (*Softassign* [10] and *Pose from Orthography and Scaling with ITerations*). *SoftPOSIT* is an iterative point correspondence scheme minimizing a global energy function based on the 2D and 3D projection points differences. There has been several implementations of the *SoftPOSIT* method in terrestrial applications. For example, Jager *et al*. [11] used *SoftPOSIT* to determine the pose of a ground vehicle based on thermal camera images, and Diaz and Abderrahim [12] used it to estimate the pose of a spinning spacecraft model. Once converged, *SoftPOSIT* can produce accurate pose estimation results with low computation resources; however, some shortcomings of the algorithm include local minimum trapping and iteration control parameter to correspondence compatibility. This paper proposes several techniques to address the above mentioned issues. First, a novel checking criteria is introduced to find the best initialization orientation. A proper initialization minimizes the chances for the optimization to converge into a local minimum. Second, two strategies are proposed on initializing the iteration control parameter by using the trace of the correspondence distance, and by approximating target object center of geometry with image centroid. Both strategies eliminates non-viable iteration control parameter prior to match optimization. Finally, optimization reset conditions will be discussed for practical considerations to numerical anomalies.

This paper is organized as follows, Sec. II formulates the *SoftPOSIT* methodology and provides definitions for frames and algorithm parameters. Sec. III provides the enhancement formulations for global minimum search and strategies for iteration control parameter selection. This includes the derivation of the centroid matching technique. Sec. IV provides descriptions of models and ICs used in the algorithm validation. Sec. V provides results and discussions of the simulation findings. Finally, Sec. VI concludes the study by comparing the enhancements to the original *SoftPOSIT* algorithm.

II. SoftPOSIT Formulation

This section provides problem definition and formulates the *SoftPOSIT* [8] methodology. Define a tracker body $\mathcal{F}_{SB}$ equipped with a single camera $\mathcal{F}_{VW}$ pointed towards a target object $\mathcal{F}_{CB}$. The frames $\mathcal{F}_{SB}$ and $\mathcal{F}_{CB}$ are located at the object Center of Geometry (COG). The camera frame $\mathcal{F}_{VW}$ has its z axis pointed outwards from the boresight of the

978-1-5090-1866-6/16/$31.00 ©2016 IEEE

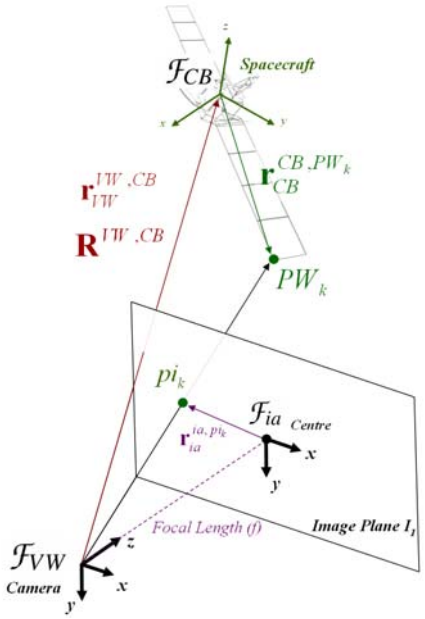


Fig. 1. Camera coordinate system definition.

camera, with the y axis pointed vertically downwards and x completing the right-handed frame. The frame $\mathcal{F}_{ia}$ is centered on the camera image with its x and y axes being parallel to $\mathcal{F}_{VW}$. Figure 1 shows the various coordinate systems considered. Let PW_k be the k^{th} point on the target that is projected on the image plane. This projected point is denoted as pi_k. The total number of points from the target model is M. These points are the corner vertices of the target body. The objective is to determine the pose, *i.e.* , $\mathbf{T}$ and $\mathbf{R}$ of the target with respect to the camera defined as follows,

$$\mathbf{T} = \mathbf{r}_{VW}^{VW,CB} = \begin{bmatrix} T_x & T_y & T_z \end{bmatrix}^T \tag{1}$$

$$\mathbf{R} = \mathbf{R}^{VW,CB} = \begin{bmatrix} \mathbf{R}_x^T \\ \mathbf{R}_y^T \\ \mathbf{R}_z^T \end{bmatrix} \tag{2}$$

where $\mathbf{T}$ is the position of $\mathcal{F}_{CB}$ w.r.t. $\mathcal{F}_{VW}$ expressed in $\mathcal{F}_{VW}$ and $\mathbf{R}$ is the rotation matrix rotating $\mathcal{F}_{CB}$ to $\mathcal{F}_{VW}$, and $\mathbf{R}_x^T$, $\mathbf{R}_y^T$, and $\mathbf{R}_z^T$ are row matrices of $\mathbf{R}$. These are also the $\mathcal{F}_{VW}$ unit vectors expressed in $\mathcal{F}_{CB}$. Let N be the number of detected image points, and pi_j denoting the j^{th} image point. The position of PW_k w.r.t. $\mathcal{F}_{VW}$ expressed in $\mathcal{F}_{VW}$ and the position of pi_k w.r.t. $\mathcal{F}_{ia}$ is denoted by $\mathbf{r}_{VW}^{VW,PW_k}$ and $\mathbf{r}_{ia}^{ia,pi_j} = \begin{bmatrix} x_j & y_j \end{bmatrix}^T$ respectively, and their relationships to the target pose is:

$$\mathbf{r}_{VW}^{VW,PW_k} = \mathbf{r}_{VW}^{VW,CB} + \mathbf{R}^{VW,CB}\mathbf{r}_{CB}^{CB,PW_k} \tag{3}$$

$$\mathbf{r}_{ia}^{ia,pi_k} = \frac{f}{Z_C}\boldsymbol{\Upsilon}\mathbf{r}_{VW}^{VW,PW_k} \tag{4}$$

where $\boldsymbol{\Upsilon} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \end{bmatrix}$ and $\mathbf{1}$ is an 2×2 identity matrix and $\mathbf{0}$ is a 2×1 zero matrix and Z_C is the z component of r_{VW}^{VW,PW_k}. Let define the *scaling ratio* s as ${}^{f}/_{T_z}$ and the *prospective ratio* w_k as ${}^{Z_C}/_{T_z}$, such that the prospective scaling term from Eq. (4) ${}^{f}/_{Z_C}$ can be replaced with ${}^{s}/_{w_k}$. If the object is far away from the camera and *field of view* (FOV) is not large, then w_k is nearly one. Evaluating Eq. (3), w_k can be written as

$$w_k = \frac{\mathbf{R}_z^T\mathbf{r}_{CB}^{CB,PW_k}}{T_z} + 1 \tag{5}$$

Let define the pose matrix $\mathbf{Q}_x$, $\mathbf{Q}_y$, and model point matrix $\mathbf{P}_k$ as, $\mathbf{Q}_x \triangleq s\begin{bmatrix} \mathbf{R}_x^T & T_x \end{bmatrix}^T$, $\mathbf{Q}_y \triangleq s\begin{bmatrix} \mathbf{R}_y^T & T_y \end{bmatrix}^T$, and $\widehat{\mathbf{P}}_k = \begin{bmatrix} \left(\mathbf{r}_{CB}^{CB,PW_k}\right)^T & 1 \end{bmatrix}^T$. Since PW_k will be determined from an existing model, M and N are in practice rarely equal. The distance between the projected model points and the camera points is

$$d_{jk}^2 = (\mathbf{Q}_x^T\widehat{\mathbf{P}}_k - w_k x_j)^2 + (\mathbf{Q}_y^T\widehat{\mathbf{P}}_k - w_k y_j)^2 \tag{6}$$

A *Global Objective Function* is formulated as:

$$E = \sum_{j=1}^{N}\sum_{k=1}^{M} m_{jk}(d_{jk}^2 - \alpha) \tag{7}$$

For a maximum correspondence between image and model points, the partial derivative of E with respect to the pose matrices is zero. Hence,

$$\mathbf{L} = \sum_{j=1}^{N}\sum_{k=1}^{M} m_{jk}\widehat{\mathbf{P}}_k\widehat{\mathbf{P}}_k^T \tag{8}$$

$$\begin{aligned} \mathbf{Q}_x &= \mathbf{L}^{-1}\sum_{j=1}^{N}\sum_{k=1}^{M} m_{jk}w_k x_j\widehat{\mathbf{P}}_k \\ \mathbf{Q}_y &= \mathbf{L}^{-1}\sum_{j=1}^{N}\sum_{k=1}^{M} m_{jk}w_k y_j\widehat{\mathbf{P}}_k \end{aligned} \tag{9}$$

where the weights m_{jk} are computed at every step based on a iteration control parameter β as follows,

$$m_{jk} = \gamma\exp(-\beta(d_{jk}^2 - \alpha)) \tag{10}$$

where γ is a normalization factor and α is to allow amplification of the d_{jk} distance. The corresponding distance between image and model points can be grouped in a matrix format as following:

$$\mathbf{D} = \begin{bmatrix} d_{1,1}^2 & \cdots & d_{1,M}^2 \\ \vdots & \ddots & \vdots \\ d_{N,1}^2 & \cdots & d_{N,M}^2 \end{bmatrix} \tag{11}$$

Once the pose matrices are computed, x and y rows of the rotation matrix axis can be found using the Singular Value Decomposition (SVD),

$$\begin{bmatrix} \frac{\mathbf{R}_x}{T_z} & \frac{\mathbf{R}_y}{T_z} \end{bmatrix} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \tag{12}$$

the rotation axis and T_z can be determined by

$$\begin{bmatrix} \mathbf{R}_x & \mathbf{R}_y \end{bmatrix} = \mathbf{U}\boldsymbol{\Upsilon}^T\mathbf{V}^T \tag{13}$$

$$T_z = \frac{2}{\Sigma_{1,1} + \Sigma_{2,2}} \tag{14}$$

Finally, $\mathbf{R}_z$ can be computed from the cross product between $\mathbf{R}_x$ and $\mathbf{R}_y$. The algorithm is terminated if the estimated pose converges below a user-defined tolerance.

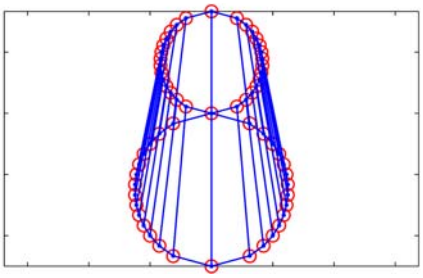


Fig. 2. 30-Point Cylinder Successful Pose Estimation. (red circles-image points, blue points-final estimated model points).

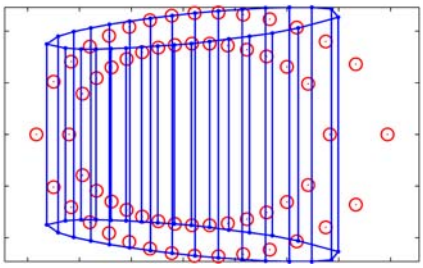


Fig. 3. 30-Point Cylinder Local Minimum Pose Estimation Failure. (red circles-image points, blue points-final estimated model points).

III. Enhancement Formulations

This section provides formulations for enhancements to the original *SoftPOSIT* algorithm. These enhancements include methods for obtaining global minimum search, iteration control parameter initialization by correspondence and by centroid matching.

A. Global Minimum Search

A major decrease in the *SoftPOSIT* estimation accuracy is caused by local minimum trapping [8], [11], [12]. The local minimum trap occurs when the iterative solution is driven by the annealing [10] and Sinkhorn's normalization process [13] to a false pose exhibiting minimum correspondences. In this state, further iterations cannot force the model pose away from the local minimum. An example of the local minimum trapping can be demonstrated in the pose estimation of a 30-points cylinder. Fig. 2 shows a successful pose estimation where as Fig. 3 demonstrates that with a different set of IC misalignment, the original algorithm converges to an incorrect local minimum solution. To solve this inherent problem of SoftPOSIT, two observations can be made on the convergence characteristics. First, in most instances, majority of pose error is removed during the first several steps of the optimization process, this can be attributed to the exponential term in Eq. (10). Therefore, one may try several different initial orientations for only a few steps before selecting the final path to optimize. Second, after the same number of iteration steps, the global minimum solution should have a lower correspondence than all local minimum solutions. As the result of the first observation, an initial condition switching logic in the pre-computation phase shall reset the initial orientation to a group of saved directions. This can be done over an angle ϕ about an axis $\vec{a}_i$, where i range from 1 to n. The angle ϕ needs to be large enough to bring the solution away from the local minimum. For the purpose of this study, ϕ is set to 90 degrees, and $\vec{a}_i$ was selected in 4 directions $(1,0,0)$, $(0,1,0)$, $(0,0,1)$, and $(1,1,1)$ relative to the initial orientation. Next, the second observation can be summarized as *the smallest maximum of the closest image to model correspondence for all n initial orientations is the likely path towards the global minimum*. This statement can be evaluated by the following steps. First, write the correspondence matrix from Eq. (11) as an array of column matrices $\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \dots & \mathbf{D}_M \end{bmatrix}$, these columns represent the correspondence from all the detected image points to one model projected point. The minimum value in the k^{th} column represent the closest image to model correspondence for the k^{th} model point, that is,

$$\hat{D}_{ki} = \underset{j \in N}{\operatorname{argmin}} \{D_{jki}\} \tag{15}$$

The maximum of these represent the *goodness* as the result from optimizing from the i^{th} initial orientation.

$$\check{D}_i = \underset{k \in M}{\operatorname{argmax}} \{\hat{D}_{ki}\} \tag{16}$$

Finally, the minimum $\check{D}_i$ signals the most likely path to the global minimum solution, as,

$$\hat{D}_g \triangleq \underset{i \in n}{\operatorname{argmin}} \{\check{D}_i\} \tag{17}$$

In the spirit of annealing optimization, this pre-computation phase is referred to as *preheating*.

B. Control Parameter based on Correspondence

From Eq. (10), the iteration control parameter β is used to control the annealing process. When β is low, the weighting will be high and vise-versa. The determination for β is largely based on trials and errors. David *et al.* [8] used a β of 0.0004 while Gold *et al.* [10] used 0.00091. Selecting the iteration control parameter can be difficult as improper matching between β and the correspondence matrix $\mathbf{D}$ may lead to numerical anomalies. In view of this, a simple β initialization is herein proposed as,

$$\beta_0 = F \frac{(M+N)/2}{\operatorname{tr}(\mathbf{D})} \tag{18}$$

where F is some scaling constant set to 2 in this study, and $\operatorname{tr}(\mathbf{D})$ is the trace of the $\mathbf{D}$ matrix. Since the trace of a matrix is also the sum of its eigenvalues, Eq. (18) allows for better compatibility between β_0 and the correspondence distances.

Eq. (18) then provides a convenient mean to select a new iteration control parameter which is useful incase of an optimization restart. In some instances, the current optimization path leads to a local minimum or divergence. For instance, if the optimization weights in Eq. (10) results in an ill-conditioned $\mathbf{L}$ matrix, which is non-inversible. In other cases, the optimization diverges, pushing the target body further away in the boresight direction while attempting to minimize the $x-y$ differences. When these numerical conditions occur, it is necessary to restart the algorithm and recompute β_0 using Eq. (18).

C. Control Parameter based on Centroid Matching

Equation (10) is a variation of the *Simulated Annealing* (SA) numerical optimization method, [14] where β is analogous to the inverse of the annealing temperature. The exponential term in Eq. (10) requires an appropriate initial temperature to be selected in order to prevent the weighting m_{jk} from becoming zero or infinity due to numerical issues. The proposed method produces a iteration control parameter β that is compatible with the point correspondence, and eliminates the need for tedious trial and error tuning campaigns of β. Specifically, it uses the image point cloud centroid to estimate the target COG. For spacecraft applications where the background is normally black space, the point cloud centroid is a good estimate of the target COG. The target image point cloud centroid is computed as,

$$\bar{\mathbf{r}}_{ia}^{ia,pi} = \sum_{j=1}^{N} \frac{r_{ia}^{ia,pi_j}}{N} \tag{19}$$

Combining Eq. (4), the scaling ratio, and Eq. (19) results in,

$$\frac{\bar{\mathbf{r}}_{ia}^{ia,pi}}{f} = \boldsymbol{\Upsilon} \frac{\bar{\mathbf{r}}_{VW}^{VW,PW}}{\overline{Z}_c} \tag{20}$$

where $\overline{Z}_c$ is the z component of $\bar{\mathbf{r}}_{VW}^{VW,PW}$. Allow the image centroid to be the target COG

$$\bar{\mathbf{r}}_{VW}^{VW,PW} \triangleq \mathbf{r}_{VW}^{VW,CB} \tag{21}$$

combining Eqs. (1), (20), and (21), the x and y COG position in the image plane is,

$$\frac{\bar{\mathbf{r}}_{ia}^{ia,pi}}{f} \triangleq \begin{bmatrix} T_x/T_z \\ T_y/T_z \end{bmatrix} \tag{22}$$

Let $\boldsymbol{\Gamma}$ be the inverse of the matrix $\mathbf{L}$, such that

$$\boldsymbol{\Gamma} = \mathbf{L}^{-1} = \begin{bmatrix} \boldsymbol{\Gamma}_{13} \\ \boldsymbol{\Gamma}_4 \end{bmatrix} \tag{23}$$

where $\boldsymbol{\Gamma}_{13}$ and $\boldsymbol{\Gamma}_4$ is the first three rows and the fourth row of the inverse $\mathbf{L}$ matrix. Combine the scaling ratio and the pose matrices gives

$$\begin{aligned} \begin{bmatrix} \mathbf{Q}_x/f \\ \mathbf{Q}_y/f \end{bmatrix} &= \begin{bmatrix} \mathbf{R}_x^T/T_z & T_x/T_z & \mathbf{R}_y^T/T_z & T_y/T_z \end{bmatrix}^T \\ &= \begin{bmatrix} \begin{bmatrix} \boldsymbol{\Gamma}_{13} \\ \boldsymbol{\Gamma}_4 \end{bmatrix} \sum_{j=1}^{N}\sum_{k=1}^{M} m_{jk} w_k \frac{x_j}{f} \widehat{\mathbf{P}}_k \\ \begin{bmatrix} \boldsymbol{\Gamma}_{13} \\ \boldsymbol{\Gamma}_4 \end{bmatrix} \sum_{j=1}^{N}\sum_{k=1}^{M} m_{jk} w_k \frac{y_j}{f} \widehat{\mathbf{P}}_k \end{bmatrix} \end{aligned} \tag{24}$$

To rewrite equations in matrix format, let

$$\begin{bmatrix} \mathbf{U}/f \\ \mathbf{V}/f \end{bmatrix} = \begin{bmatrix} \frac{x_1}{f} & \cdots & \frac{x_N}{f} \\ \frac{y_1}{f} & \cdots & \frac{y_N}{f} \end{bmatrix} \tag{25}$$

$$\mathbf{M} = \begin{bmatrix} m_{11} & \cdots & m_{1M} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NM} \end{bmatrix} \tag{26}$$

$$\mathbf{P}_M = \begin{bmatrix} \widehat{\mathbf{P}}_1 & \cdots & \widehat{\mathbf{P}}_M \end{bmatrix}^T \tag{27}$$

such that w_k, for all image points, can be grouped into a matrix, by combining Eq. (27) and Eq. (5), as follows

$$\mathbf{w}_M = \begin{bmatrix} w_1 & \cdots & w_M \end{bmatrix}^T = \mathbf{P}_M \begin{bmatrix} \mathbf{R}_z/T_z \\ 1 \end{bmatrix} \tag{28}$$

Also, let

$$h_k = w_k \boldsymbol{\Gamma}_4 \widehat{\mathbf{P}}_k \tag{29}$$

$$\mathbf{h}_M = \begin{bmatrix} h_1 & \cdots & h_M \end{bmatrix}^T = \mathbf{w}_m \circ \left(\mathbf{P}_M \boldsymbol{\Gamma}_4^T \right) \tag{30}$$

where $\mathbf{A} \circ \mathbf{B}$ is the Hadamard product between matrix $\mathbf{A}$ and $\mathbf{B}$. Extract T_x and T_y from Eq. (24), substitute in Eqs. (25)-(30), and rewrite their ratio with T_z as

$$\begin{bmatrix} T_x/T_z \\ T_y/T_z \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{N} \frac{x_j}{f} \sum_{k=1}^{M} m_{jk} h_k \\ \sum_{j=1}^{N} \frac{y_j}{f} \sum_{k=1}^{M} m_{jk} h_k \end{bmatrix} = \begin{bmatrix} \mathbf{U}/f \\ \mathbf{V}/f \end{bmatrix} \mathbf{M} \mathbf{h}_M \tag{31}$$

The weighting matrix is defined by Eq. (10), let

$$\mathbf{D}_3 = \begin{bmatrix} \mathbf{U}/f \\ \mathbf{V}/f \end{bmatrix} \gamma \exp\left(-\beta_0 (\mathbf{D} - \alpha \mathbf{1})\right) \tag{32}$$

where $\mathbf{1}$ is a $N \times M$ matrix of ones. Combine Eqs. (22), (31), and (32) results in,

$$\frac{\bar{\mathbf{r}}_{ia}^{ia,pi}}{f} \triangleq \mathbf{D}_3 \mathbf{h}_M \tag{33}$$

let define the left-hand side term $\bar{\mathbf{r}}_{ia}^{ia,pi}/f$ as $\mathbf{r}$, such that its square becomes

$$\mathbf{r}^T \mathbf{r} \triangleq \mathbf{h}_M^T \mathbf{D}_3^T \mathbf{D}_3 \mathbf{h}_M \tag{34}$$

Let rewrite $\mathbf{r}^T \mathbf{r}$ as r_2, and $\mathbf{D}_3^T \mathbf{D}_3$ as $\mathbf{D}_4$ such that Eq. (34) can be rewritten as

$$r_2 \triangleq \mathbf{h}_M^T \mathbf{D}_4 \mathbf{h}_M \tag{35}$$

To solve Eq. (35), let

$$\mathfrak{F} \triangleq \mathbf{h}_M^T \mathbf{D}_4 \mathbf{h}_M - r_2 = 0 \tag{36}$$

The discrete rate of change of $\mathfrak{F}$ with respect to β is

$$\mathfrak{F}'_{n-1} = \frac{\mathfrak{F}_n - \mathfrak{F}_{n-1}}{\beta_n - \beta_{n-1}} \tag{37}$$

Using Newton's method of gradient descent, the initial iteration control parameter is computed as

$$\beta_{n+1} = \beta n - \frac{\mathfrak{F}_n}{\mathfrak{F}'_{n-1}} \tag{38}$$

Typically, β_0 can converge to 10^{-14} accuracy within 30 iterations. For cases where the Newton's method cannot produce the root value, it is appropriate to compute the initial iteration control parameter β_0 from the average trace of the $\mathbf{D}$ matrix as previously obtained by Eq. (18).

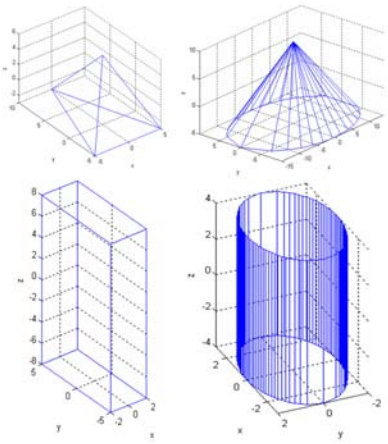


Fig. 4. Elementary building shapes.

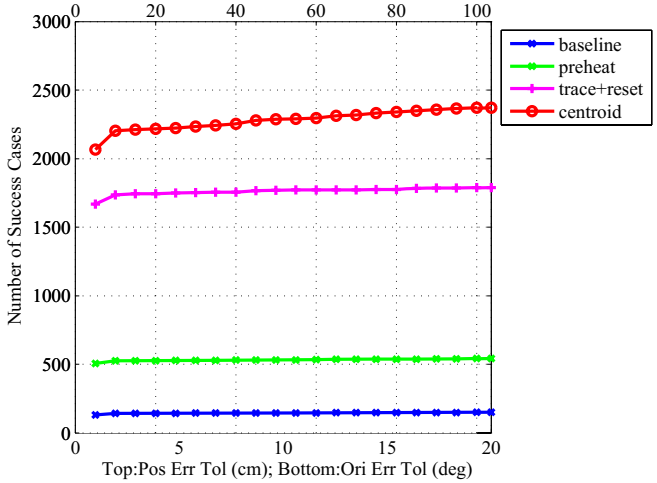


Fig. 5. Number of Successes as a function of Error Tolerances.

IV. Simulation Setup

To test the proposed enhancements, four batches of $6,545$ cases were simulated, for a total of $26,180$ simulations. The 4 batches represent incremental additions of the proposed enhancements. The batch orders are: baseline, *preheating*, β_0 by trace with error reset with z-limiting, and β_0 by COG. Within each batch, the number of cases is the result of combinations in object shapes, model points, model initial poses, and target object initial poses. The internal model consists of stored vertices of elementary shapes such as rectangles, cylinders, ellipses and cones (see Fig. 4). These elementary shapes can be used to construct more complex objects such as spacecraft and space stations. Eleven cases of 5 elementary shapes with points varying between 3 to 30 are specified to be within a $2 \times 2 \times 2$ meter envelope. The termination β is $1,000$, the incrementing rate for β is 1.05 [8], and Sinkhorn normalization iteration was set to 100 cycles. The model initial condition was varied in 5 directions, these are combinations of translation and Euler angle misalignments. Pose spans ± 10 meters and ± 135 degrees on all five axis with the exception of the camera boresight. The target IC relative to $\mathcal{F}_{VW}$ was varied in seven translational directions and 17 orientations with maximum magnitude of ± 5 meters and ± 90 degrees. This represents a hypercube of 119 directions. Combination of the target and model IC and the 11 models cases produces $6,545$ simulation cases under one batch run.

V. Simulation Results

The SA optimization contains many local minimums for the pose solution to become trapped. One solution is by using restarts based on a percentage match between the image point correspondence to the model. On each restart, the initial condition is changed randomly to ensure a different path is taken. However, this method can be costly, the mean number of restarts for finding a good pose can range between 100 to $1,000$ and the time for finding a good pose is on the order of 4 seconds to 6 minutes [a] for targets with 20 to 30 points. [8] Setting the minimum pose success criteria to 5 cm and 1 degree, which is roughly 1 percent of the IC misalignment. Without any restarts, the baseline success rate for a single batch is 131 out of $6,454$ cases with a mean time of just less than 3 seconds [*]. This is primarily due to the results falling into the local minimum traps. The objective of this work is to increase the pose estimation accuracy but still keep optimization time low enough for near real-time operations. To test the proposed enhancements, the four batches of tests were ran each with accumulating enhancements added to it. Success is evaluated by comparing the pose error to the ground truth and the acceptable pose tolerance. With changes in acceptable pose tolerance, the success rate also differs, Figure 5 shows the successful cases as the pose tolerance increased from 1 to 20 degrees and 5 cm to 1 meter. For all enhancements with the exception of the β_0 initialization by COG, the success rate remain relatively constant with increasing error tolerance. The β_0 initialization by COG show a steady increase as error tolerance increased. Considering the minimum tolerance case, the *preheating* global minimization search produced a pose success increase of 507 cases. With β_0 initialization by correspondence matrix trace and reset due to numerical anomaly, the pose success increased to $1,667$ cases. Finally, there was $2,067$ successful cases with β_0 COG initialization. It should be noted the basic shapes tested are symmetrical and in some cases circular, which may result in ambiguous solutions. However, a real world target vehicle given its geometric details will most likely not be symmetrical and therefore resolving the pose ambiguities. Figure 6 shows the average match time as a percentage of the of the baseline algorithm. The results show almost 50 percent time increase in each enhancement with the β_0 COG initialization time increasing by 75 percent.

Figure 7 shows the enhanced *SoftPOSIT* estimation of a Radarsat model image captured by a monocular infrared (IR) camera. The position initial condition for the internal model

[a] Simulated in Linux Matlab with 2.4GHz Pentium 4 processor [8]

[*] Simulated in 32bit-Windows Matlab with 2.4GHz Intel® Core™ 2 Quad Q6600 processor

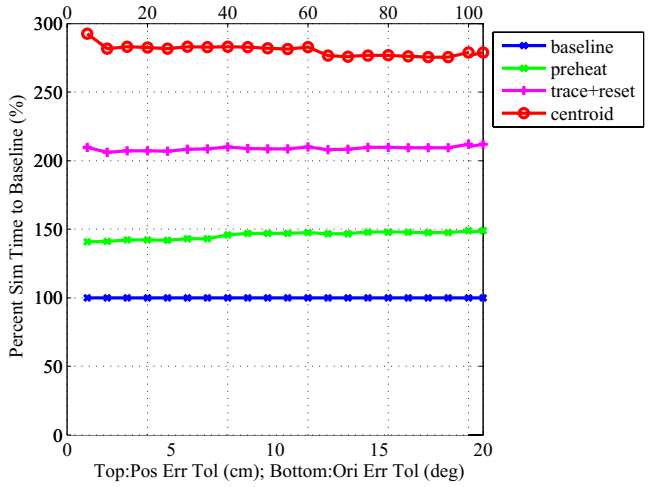
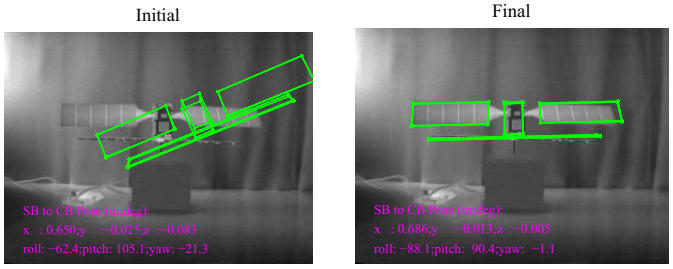


Fig. 6. Percent Iteration Time as a function of Error Tolerance.



(a) pre-match (b) post-match

Fig. 7. *SoftPOSIT* pose estimation of a Radarsat model captured by an monocular infrared camera.

is 0.650, -0.025, and -0.083 meters for x, y, and z axis respectively, from the $\mathcal{F}_{SB}$ frame to the $\mathcal{F}_{CB}$ frame expressed in the $\mathcal{F}_{SB}$ frame. The orientation initial condition for the internal model is -62.4, 105.1, and -21.3 degrees for $roll$, $pitch$, and yaw respectively. The orientation angles are in the p-y-r Euler angle rotation sequence, rotating from the $\mathcal{F}_{SB}$ frame to the $\mathcal{F}_{CB}$ frame. For the IR image in Figure 7, the enhanced softPOSIT iteration took 1.031 seconds to complete 43 iterations. The final target model satellite pose is 0.686, -0.013, -0.005 meters and -88.1, 90.4, -1.1 degrees. In this case, the initial misalignment of the internal model is large, this will increase the complexity of the iteration process and lengthens the optimization time. Large misalignments will have more chances of falling into local minimum traps as they must traverse across a large path before settling into the global optimal path. Consider the misalignment case over one single video frame. The frame rate update is typically between one to 30 hertz. Given the rotational rate of a tumbling target satellite is on the order of $1-2$ deg/s [15]. Assuming 1 second frame update rate, this is roughly 10 times lower than the initial misalignment of the test example. When the initial misalignment is on the order of 2 degrees, the same test case reaches a solution in 0.873 seconds.

VI. Conclusions

In conclusion, enhancements made to the *SoftPOSIT* algorithm include global optimization search, iteration control parameter initialization by trace of the correspondence matrix, and by COG. The goal of the enhancements were to increase pose estimation accuracy while maintaining computation speed. Results show a significant accuracy increase w.r.t. the baseline with relatively low increase in computation time. It should be noted however there is still much room to improve the algorithm to gain acceptable reliability. Future work include investigation of efficient ways to restart the algorithm based on recursive learning.

Acknowledgements

This research was funded by the Natural Sciences and Research Council of Canada's Alexander Graham Bell Canada Graduate Scholarship CGSD3-453738-2014.

References

[1] L. Castellani, S. Llorente, M. Fernandez, J.M. Ruiz, A. Mestreau-Garreau, A. Cropp, and Santovincenzo, "Proba-3 mission," *Space Science and Engineering, Intl. Journal of*, vol. 1, no. 4, pp. 349–366, 2013.

[2] S. Sharma and S. D'Amico, "Comparative assessment of techniques for initial pose estimation using monocular vision," *Acta Astronautica*, 2016.

[3] C. Doignon, "An introduction to model-based pose estimation and 3-d tracking technique," in *Scene Reconstruction Pose Estimation and Tracking*, Bellingham WA, 2007, rustam Stolkin (Ed.).

[4] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Graphics and Image Processing*, vol. 24, no. 6, pp. 381–395, 1981.

[5] P. Besl and N. McKay, "A method for registration of 3-d shapes," *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, vol. 14, no. 2, pp. 239–256, 1992.

[6] S. D'Amico, M. Benn, and JNørgensen, "Pose estimation of an uncooperative spacecraft from actual space imagery," *Space Science and Engineering 5, Intl. Journal of*, vol. 2, no. 2, pp. 171–189, 2014.

[7] V. Lepetit, "Epnp: an accurate o(n) solution to the pnp problem," *Computer Vision, Intl. Journal of*, vol. 81, no. 2, pp. 155–166, 2009.

[8] P. David, D. Dementhon, R. Duraiswami, and H. Samet, "Softposit: simultaneous pose and correspondence determination," *Computer Vision, Intl. Journal of*, vol. 59, no. 3, pp. 259–289, 2004.

[9] D. Dementhon and L. Davis, "Model-based object pose in 25 lines of code," *Computer Vision, Intl. Journal of*, vol. 15, no. 1, pp. 123–141, 1995.

[10] S. Gold, A. Rangarajan, C. P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2d and 3d point matching: pose estimation and correspondence," *Pattern Recognition*, vol. 31, no. 8, pp. 1019–1031, 1998.

[11] K. Jager, M. Hebel, and K. Bers, "Automatic 3d object pose estimation in ir image sequences for forward motion application," in *Automatic Target Recognition XIV, Proc. of SPIE*, vol. 5426, 2004.

[12] J. Diaz and M. Abderrahim, "Modified softposit algorithm for 3d visual tracking," in *IEEE International Symp. on Intelligent Signal Processing*, Oct. 2007.

[13] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The Annals of Mathematical Statistics*, vol. 35, no. 2, pp. 876–879, 1964.

[14] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science, New Series*, vol. 220, no. 4598, pp. 671–680, 1983.

[15] D. Kucharski, G. Kirchner, F. Koidl, R. Fan C., Carman, C. Moore, A. Dmytrotsa, M. Ploer, G. Bianco, M. Medvedskij, A. Makeyev, G. Appleby, M. Suzuki, J. Torre, Z. Zhang, L. Grunwaldt, and F. Qu, "Attitude and spin period of space debris envisat measured by satellite laser ranging," *Geoscience and Remote Sensing, IEEE Trans. on*, vol. 52, no. 12, pp. 7651–7657, 2014.