Satellite articulation characterization from an image trajectory matrix using optimization*

David H. Curtis

Air Force Institute of Technology, Wright-Patterson AFB, Ohio

Richard G. Cobb

Air Force Institute of Technology, Wright-Patterson AFB, Ohio

CONFERENCE PAPER

Autonomous on-orbit satellite servicing and inspection benefits from an inspector satellite that can autonomously gain as much information as possible about the primary satellite. This includes performance of articulated objects such as solar arrays, antennas, and sensors. This paper presents a method of characterizing the articulation of a satellite using resolved monocular imagery. A simulated point cloud representing a nominal satellite with articulating solar panels and a complex articulating appendage is developed and projected to the image coordinates that would be seen from an inspector following a given inspection route. A method is developed to analyze the resulting image trajectory matrix. The developed method takes advantage of the fact that the route of the inspector satellite is known to assist in the segmentation of the points into different rigid bodies, the creation of the 3D point cloud, and the identification of the articulation parameters. Once the point cloud and the articulation parameters are calculated, they can be compared to the known truth. The error in the calculated point cloud is determined as well as the difference between the true workspace of the satellite and the calculated workspace. These metrics can be used to compare the quality of various inspection routes for characterizing the satellite and its articulation.

1 INTRODUCTION

Due to the nature of the space environment, direct human interaction with on-orbit satellites is incredibly dangerous and costly. Therefore, with the exception of a few manned missions, interaction with satellites is traditionally limited to windows of radio communications and observations from telescopes. These methods are inadequate for inspection and monitoring of a satellite to determine performance. To perform these monitoring tasks, an inspector satellite in close proximity to the primary satellite could use sensors, such as a camera, to characterize the primary satellite and verify the performance of articulated objects such as sensors, solar arrays, robotic arms, and communications antennas. Imagery contains a lot of information, however it is also difficult to transmit to the ground due to the large size of video files. This motivates the creation of algorithms that can create a model that characterizes the performance of the target satellite autonomously using computer vision. Developing autonomous methods is also important for autonomous repair missions[9] or deep space missions that have little or no ability to communicate with human operators on Earth.

*The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. This document has been cleared for public release.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

1.1 Background

Research in the area of computer vision for proximity operations is very diverse, ranging from stereo vision used to estimate target satellite moment of inertia (MOI)[24] to demonstration of fully autonomous rendezvous and capture.[9] Some of these algorithms rely upon markers on the primary satellite that assist the computer vision algorithm in determining pose.[8, 23] Others rely on prior knowledge of the primary satellites configuration.[7, 13] Some methods rely on stereo vision systems,[4, 24] some on monocular vision systems,[7, 10, 16] and some use laser illumination of reflective markers.[9] They all use a computer vision method that identifies features in images and matches those features from frame to frame (or in corresponding frames in the case of stereo systems). The relative position of the features is then estimated using some type of estimation filter such as an extended Kalman filter (EKF) or a particle filter. The attitude of the primary satellite is then estimated by using feature points to define a primary reference frame,[16, 28] or by using a known model of the primary satellite.[13] Alternatively, if the orientation of feature points is known, the attitude can be calculated directly and used as the measurement in the EKF.[16, 23, 28] Feature points can also be used to create a 3D model of the satellite.[6] Apart from their own previous work[2] the authors have not discovered previous research attempting to use computer vision to determine if the primary satellite is in-fact a rigid body, or if it is undergoing some type of articulation.

While articulation has not been investigated for space applications specifically, there has been extensive work on identifying and characterizing articulation in the computer vision field. The seminal works of Yan and Pollefeys [26] and Trasadern and Reid [22] extend the concepts of structure from motion through factorization of a trajectory matrix [21] from rigid bodies only to multiple linked rigid bodies. Numerous researchers have further extended these concepts to characterize articulated motion.

Yucer et al.[29] developed a method of reconstructing an object with multiple articulated motions using optimization techniques. First, a 'ray-space optimization' method is used to convert each 2D image trajectory into a 3D trajectory. This method requires the camera motion *a priori*, however this requirement is not necessarily restraining in an application in which the relative trajectory between the inspector and the primary satellite is known. The 3D trajectories of each point are then used to segment the motion into N rigid bodies. Once segmented, an optimization routine is used to find a rotation matrix (R_f^t), shape (Ω_n), and translation (T_f^n) for each rigid body that minimizes the error in reprojecting the points to image coordinates (W_f^n). Next, the kinematic chain is estimated and another optimization is done with the articulation constraints enforced.

Paladini et al.[15] present a method in which the motion matrix (identifying the motion of the camera) and shape matrix (identifying the shape of the object) are solved for in an alternating fashion with least squares. First, an initial solution to the motion matrix ($M^{(0)}$) is found using the method outline by Tresadern and Reid[22]. This is then refined by optimizing a cost function that includes the metric constraints and the articulation constraints. They use the method in Marques and Costeira[12] to enable their method to handle missing data. Results suggest the algorithm is capable of handling trajectory matrices in which over 60% of the columns contain missing data without detriment to accuracy of the recovered shape. Zhang and Hung[30] approach the problem of recovering articulated structure from motion as an ellipsoid fitting problem. They begin by noting that each subset of the trajectory matrix representing the points on a rigid body, can be represented as a 3D ellipsoid. Since points corresponding to the same motion will lie on the same ellipsoid, this can be used to segment the points. They use the error in fitting a point to an ellipsoid as a metric with which to segment the feature points into separate motions.

Much of the literature on recovery of articulated structure from motion is focused on capturing the motion of humans. Some of this research can be applied to articulated motion of any kind. For instance, Fayad et al.[5] present a method of automatically recovering the 3D shape and structure of an articulated body. Their method uses an optimization routine to assign feature points to the motion that minimizes the total re-projection error of the solution. The method allows points to belong to more than one motion. This overlap of points on multiple motions defines the joint between the two objects and allows the kinematic chain to be built within the optimization framework. Russell et al.[18] expand on the method of Fayad et al. to include the capability to segment motion into independent objects and further segment independent objects into multiple parts based on dependent motions such as articulation.

Understanding articulated motion is also important to enable robots to learn about how objects move in order to allow the robots to manipulate the objects in the future. With this motivation, Pillai et al.[17]

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

developed a training process by which the parameters of everyday articulate objects, such as doors and drawers are determined from video of a person using those objects. The method in this work builds on existing work in the computer vision field and applies it to the inspection of a satellite in space.

2 METHOD

The method developed herein attempts to detect and characterize the articulation and shape of a satellite from a simulated trajectory matrix representing the time history of feature points that would be seen from a monocular camera on an inspector satellite. The method begins with an inspection route and the time history of a point cloud representing a nominal satellite with articulated motion. Given the inspection route and the point cloud, a trajectory matrix is created using a pinhole camera model, taking into account occlusion and shadowing. The trajectory matrix is $2F \times P$ matrix where F is the number of image frames and P is the number of feature points. The trajectory matrix and the inspector satellite trajectory are then used to characterize the satellite and its articulated motion. Fig. 1a provides an overview of the current method that is the main contribution of this paper. This method is a combination of multiple static optimization routines similar to the process outlined by Yucer et al.[29]

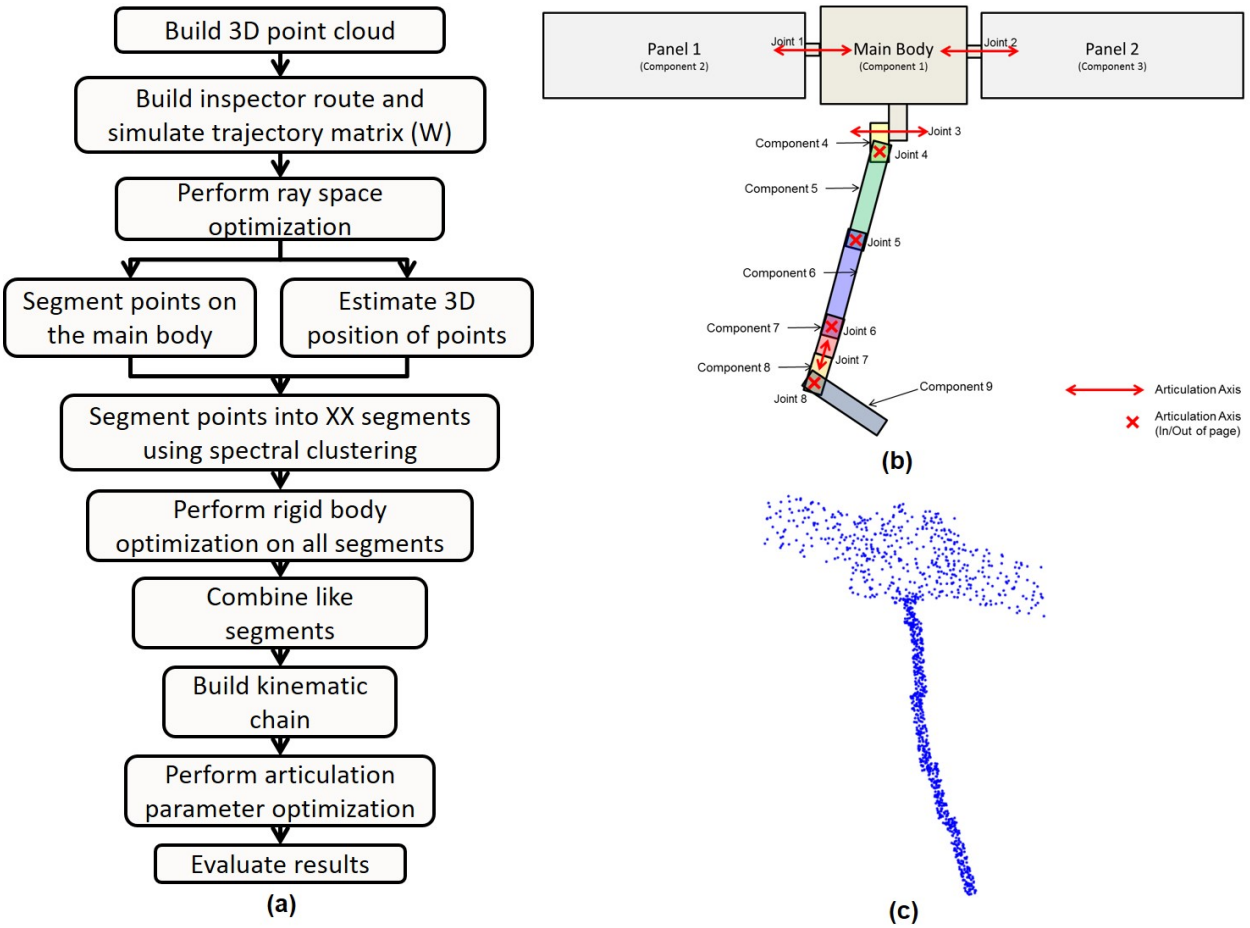

Fig. 1. a) Outlined of the proposed method. b) Diagram showing component and articulation axis naming. c) Example point cloud representation of the satellite.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

2.1 Simulation set-up

For this work a nominal satellite was developed for testing. This satellite consists of a main body, two articulating panels, and a 6-component articulating arm. Each of the joints is a revolute joint meaning that it articulates about a constant axis. Fig. 1b shows the model satellite which is approximately 2 units deep (x-direction), 10 units wide (y-direction), and 10 units tall (z-direction) with 8 articulation joints as shown. Points were randomly spread over each face to build the point cloud shown in Fig. 1c.

The primary inspection route investigated in this work is the 2x1 elliptical natural motion circumnavigation (NMC) route identified by Sabol et al.[19] The trajectory matrix is created by translating visible points from the point cloud to the camera image plane at each position in the inspection route where an image is taken. Points are visible if they are illuminated by the sun vector and they are not occluded by another part of the satellite. An error function is used to gradually fade points in and out of view as the angle between the sun or camera and the normal vector translates through 90°. Visible points are translated to the camera image plane using a pinhole camera model. Each column in the trajectory matrix contains the image coordinates (u, v) of a feature point in each image it is seen. Each set of two rows in the trajectory matrix contains the image coordinates of every point in the point cloud that is visible in that frame. The trajectory matrix is a common method of containing feature point data in the computer vision field. Note that while the nominal satellite contains panels that resemble solar arrays, no effort was taken to ensure their motion is consistent with a solar array. Both sides of the array are illuminated at different portions of the route.

2.2 Ray Space Optimization

The ray space optimization technique outlined in Yucer et al.[29] provides an excellent method of estimating 3D shape from a 2D image coordinates when the camera motion is known. The method parametrizes the 3D location of each point (p) in a given frame (i) using the camera center (C_i), the direction vector from the camera center to the point (D_i^p) and the distance along the ray from the camera center to the point (μ_i^p) using the following equation:

$$S_i^p = C_i + \mu_i^p D_i^p \tag{1}$$

where S_i^p is the 3D location of point p in frame i. Since the points are not stationary there are many valid 3D paths that can result in the same 2D image coordinates, therefore, an added assumption is made that the points track smoothly from frame to frame. This leads to a cost function that is only a function of the depth μ_i^p at each frame.

$$\min_{\vec{\mu}^p} \sum_{i=1}^{F-1} \omega_i^p \|(C_i + \mu_i^p D_i^p) - (C_{i+1} + \mu_{i+1}^p D_{i+1}^p)\|^2 \tag{2}$$

The term ω_i^p is a weighting term that is based on the 2D distance of the point in frame $i+1$ from the epipolar line corresponding to the point location in frame i. This weighting method effectively keeps the 3D location of the point when it is moving near its location when it is static. The distance from the epipolar line can also be used to determine if the point is stationary. With the assumption that the known camera motion is with respect to the main body of the satellite, points that are stationary are points on the main body (or components that are rigidly attached to the main body throughout the inspection). Points that have an average distance from the epipolar line below a threshold (γ_s) are considered stationary and are segmented to the main body.

Knowing that a point is stationary also allows the calculation of its position to be simplified to a triangulation problem that can be solved using linear least squares. For a stationary point $S_1^p = S_2^p = ... = S_F^p$, therefore equation (1) becomes $S^p = C_i + \mu_i^p D_i^p$. When written for each of the F frames there are 3F equations. Since S is no longer different for each frame there are only F+3 unknowns. This set of equations, in equation (3), can be written in the form of $Ax = b$ (equation (4)) and solved with linear least squares (I

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

is a 3×3 identity matrix).

$$\begin{aligned} S^p - \mu_1^p D_1^p &= C_1 \\ S^p - \mu_2^p D_2^p &= C_2 \\ \vdots &= \vdots \\ S^p - \mu_F^p D_F^p &= C_F \end{aligned} \tag{3}$$

$$\begin{bmatrix} I & -D_1^p & 0 & \cdots & 0 \\ I & 0 & -D_2^p & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I & 0 & 0 & \cdots & -D_F^p \end{bmatrix} \begin{bmatrix} S^p \\ \mu_1^p \\ \mu_2^p \\ \vdots \\ \mu_F^p \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_F \end{bmatrix} \tag{4}$$

Note that in this framework, points are independent, and therefore the optimization is performed on each point separately using only the frames in which the point is visible. The method of simulating the trajectory matrix allows for individual feature points to be lost and reacquired resulting in gaps where a feature point is not seen. The ray space optimization results were improved by performing the summation in equation (2) only for temporally adjacent frames.

2.3 Segment Points

One of the most challenging aspects of this problem is segmenting points into groups that represent distinct rigid bodies. The nature of a circumnavigation inspection route yields a trajectory matrix with a significant amount of missing data. Fig. 2 shows an example trajectory matrix where empty elements are shown in black and elements containing data are shown in white. Spectral clustering is a popular method of segmenting data.[25] While there are multiple methods of spectral clustering, they all operate on some type of similarity matrix. For P points, a similarity matrix is $P \times P$ in dimension with each element representing how similar the point corresponding to its column is to the point corresponding to its row. The type of similarity metric used varies widely. Some, such as LSA [27] use a metric based on the angles between the subspaces of nearest neighbors, while others use a metric based on the range and velocity between points.[30] Once the similarity metric is choosen, there are numerous ways of segmenting the data. For this work we investigated both spectral clustering using k-means [14] as well as recursive 2-way spectral clustering.[20, 29] Both methods involve solving for the eigenvectors of the Laplacian matrix. The Laplacian matrix is the similarity matrix minus a diagonal matrix of the row sums of the similarity matrix. The reader is directed to [11] for additional information on spectral clustering, and [25] for an overview of its use in motion segmentation.

Fig. 2. Example Trajectory Matrix for 1 complete NMC.

Due to the sparsity of the trajectory matrix there are many points on the same rigid component that do not share any common frames. This makes it difficult to segment them onto the same rigid component. When added to the fact that the number of rigid components is not assumed to be known *a priori*, over segmentation is necessary. The k-means clustering technique requires the number of segments (k) to be given while the 2-way spectral clustering technique used by Yucer et al. [29] only requires an estimate for the number of segments. In testing both methods produced similar results with approximately 90% of points segmented correctly. For this work a similarity matrix based on the range and 2D velocity [5] between image points was used with spectral clustering using k-means [1, 14] and a value of k ranging from 15-30 based on the number of active joints.

2.4 Rigid Body Optimization

Once the points are segmented, into rigid bodies the next step is to find how all the rigid bodies are moving. Similar to Yucer et al.[29] we sought a translation, rotation, and shape that minimized the reprojection error and a smoothness constraint.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

$$\min_{R,T,\Omega} \underbrace{\sum_{i=1}^{F} \|W_i^n - P_i(R_i^n \mathbf{\Omega}^n + \tilde{T}_i^n)\|^2}_{\text{Reprojection Error}} + \underbrace{\lambda_{rb} \sum_{i=2}^{F} \left[acos(.5(trace(R_i^n (R_{i-1}^n)^T) - 1)) + \|T_i^n - T_{i-1}^n\|\right]}_{\text{Smoothness Constraint}} \quad (5)$$

This optimization is performed for each segment. W^n are the columns in the trajectory matrix for the n-th segment. P is the camera matrix that projects points in the world frame to the camera image plane using homogeneous coordinates. Ω^n is the 3D location of the points in the segment in its body frame. R^n is the rotation matrix that rotates body frame for segment n to the world frame. $\tilde{T}^n$ is the translation from the origin of the world frame to the center of the points in Ω^n while T^n is the translation from the origin of the world frame to the origin of the body frame. They are related by: $\tilde{T}_i^n = T_i^n - R_i^n \bar{\Omega}^n$. Using $\tilde{T}^n$ in the reprojection error portion of the cost function helps to decouple the rotation from translation.

There are multiple rigid body shapes and motions that can minimize the reprojection error. Since the rigid bodies are most likely to follow smooth paths, the smoothness constraint is added to encourage solutions that follow a smooth rotational and translational paths. λ_{rb} is used to weight the smoothness constraint. For the simulations completed for this paper λ_{rb} was set to 1e-8, however that value will be dependent on the scaling of the trajectory matrix. Setting it too low will not enforce smoothness and may lead to a jittery path while setting it too high will minimize the rigid body motion at the expense of reprojection error. Note that the smoothness constraint is only applied when frames are temporally adjacent.

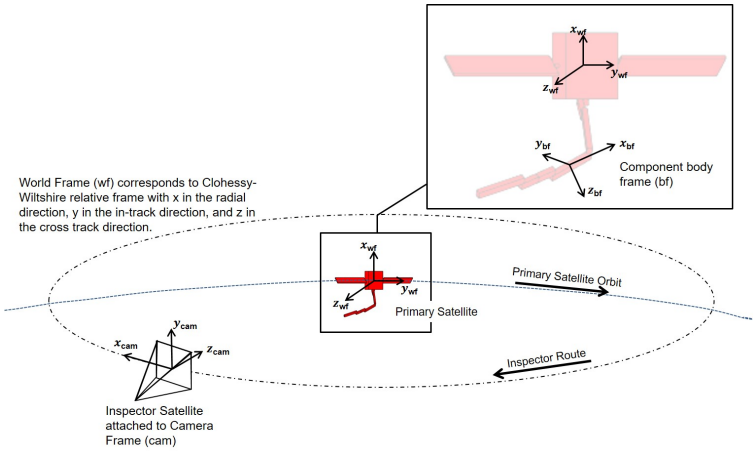

Fig. 3. Reference frames used in this work. The 'world frame' (wf) corresponds to the CW frame and moves with the primary satellite in its orbit. Each component has a 'body frame' (bf) attached to it which is arbitrarily assigned. The 'camera frame' (cam) is attached to the inspector camera with the positive z axis along the imaging axis and the positive x axis aligned with the camera's direction of travel in the world frame.

The rotation matrix (R^n) is parameterized using an Euler axis ($\vec{a}$) and an Euler angle (ϕ). The Euler axis is not constrained to be unit length, instead it is normalized before being used to calculate R_n.

This cost function is highly non-linear and contains $(7 \times F) + (3 \times P)$ optimization variables. With 100 frames and 50 points per rigid body (which is typical for simulations run) this translates to 850 optimization variables. Many optimizers, including Matlab's fminunc which was used in this work, calculate derivatives via finite differences if they are not supplied. This method was used initially, however it was computationally slow since the optimizer needs to call the function once for each optimization variable to calculate finite differences at each iteration. To speed up computation, analytical derivatives were calculated and supplied to the optimizer. The derivatives are available in Appendix A. When the analytical derivatives are supplied the computational time decreases by approximately 99.7% (the optimization was done ≈ 360 times faster).

To begin optimization, an initial guess must be supplied for all the optimization variables. The ray space optimization results provide a good method of initializing the shape (Ω^n) and the translation (T^n). Ray space optimization provides a 3D estimate for each point location in each frame (S_f^p), however this estimate is in the world frame while Ω^n is defined in the body frame. The body frame can be defined for each segment arbitrarily, therefore if there is an image frame that contains all of the points in the segment the position of the rigid body in that image frame (i) can be used to define the body frame by centering the 3D estimate for that image frame as follows: $\Omega^n = S_i^n - \bar{S}_i^n$ where $\bar{S}_i^n$ is the center of the segments 3D estimate at that image frame. If there are no image frames containing all of the points, the image frame containing the most points is used to define the body frame and the unseen points are set to zero for initialization. The translation is

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

initialized to be the center of the available 3D estimates at each frame: $T_i^n = \bar{S}_i^n$.

The rotation parameters are initialized by running 5,000 random seeds through equation (5) and selecting the one resulting in the lowest reprojection error. Each of the seeds uses shape and translation as previously discussed. The rotation matrix R_f is determined from a randomly selected constant Euler axis $\hat{a}$ and a linear set of Euler angles $\vec{\phi}$ of a random slope.

Outlier rejection is conducted every 100 iterations during rigid body optimization. Points that have a reprojection error higher than some multiplier (γ_f) times the average per point reprojection error and points that have an average range to all other points higher than γ_r standard deviations from the mean range between points are rejected as outliers.

2.5 Combine Segments

Since the data has been over segmented, the next step is to combine (merge) any segments that are part of the same rigid body. To do this, four adjacency matrices that compare the location and rotation of each segment to each other segment are created. If two segments are in fact the same rigid body, the rotations should be the same, therefore the rotation between their body frames ($R_i^{b_{n_1},b_{n_2}}$) should be constant for all common image frames.

$$R_i^{b_{n_1},b_{n_2}} = (R_i^1)^T R_i^2 \tag{6}$$

The average variance of the elements of $R_f^{b_{n_1},b_{n_2}}$ is used to define the (n_1, n_2) position of an $N \times N$ adjacency matrix where N is the number of segments.

It is assumed that segments that are part of the same rigid body are close to one another, therefore an adjacency matrix is created that compares the range between translations in common frames. Segments that are the same rigid body will also remain at the same distance from one another over common frames so a third adjacency matrix is created comparing the distance between components average location, allowing comparison of segments that do not appear in common frames. A fourth adjacency matrix is created to compare the variance of the range between the translations. These four adjacency matrices are used to determine which pairs of segments to test for merging.

For each segment, the closest two segments from each of the four adjacency matrices are selected for testing to determine if they should be merged. Rigid body optimization is performed on the combined segments as outlined in the previous section. If the function value is below 150% (γ_c) of the average function values of the segments independently, the two segments are merged and the adjacency matrices are recalculated. This is continued until all segments have been checked without triggering a merge.

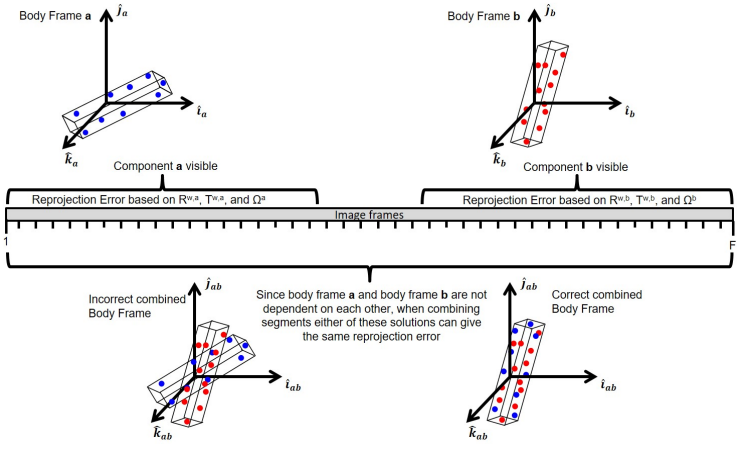

Fig. 4. Combining different segments that are not seen in the same frame is challenging because there is no information to relate one body frame to the other.

A method of combining segments without common frames was developed, however the resulting shape often was not correct. The reason for this is that there is no information linking the points to one another. For instance, the points in segment a can be defined in a body frame b_a and the points in segment b can be defined in a body frame b_b. Since there is no overlap, the rotation and translation parameters for the image frames in which a is seen can be different from the rotation and translation parameters for the image frames in which b is seen. Therefore, body frame a can be different from body frame b without consequence as illustrated in Fig. 4. This contributes to the difficulty in completely combining segments into the smallest possible number of rigid bodies.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

2.6 Build Kinematic Chain

The next step is to determine the kinematic chain that best describes which components are linked to each other. This is done by evaluating the average range adjacency matrix mentioned in the previous section. The kinematic chain is the minimum spanning tree of a graph (G) with each component as a node and edges defined by the average distance between components in common frames (CF). To ensure the root of the kinematic chain corresponds to the main body, the segments are first ordered by their average translation.

$$G(j,k) = \frac{1}{CF}\sum_{i=1}^{CF} \|T_i^j - T_i^k\| \tag{7}$$

The kinematic chain can be expressed as a table of dimension $(N-1)\times 2$ containing parent-child relationships.

$$H = \begin{bmatrix} parent_1 & child_1 \\ parent_2 & child_2 \\ \vdots & \vdots \\ parent_{N-1} & child_{N-1} \end{bmatrix} \tag{8}$$

This table is re-ordered to ensure that each segment appears as a child before it appears as a parent. Components that do not share common frames with their parent component are eliminated. Note that this method is not robust to cycles in the kinematic chain.

2.7 Articulation Parameter Optimization

Once the kinematic chain is known, the articulation constraints can be applied. For this application the joints are assumed to be revolute. Yucer et al. [29] use a similar method to the one presented here, however they assume universal joints and do not include the joint locations as optimization variables. The articulation parameters are the articulation axes between linked parts, the angle of articulation, and the joint locations. For N components, there are N-1 joints. For each joint, p identifies the parent component and c identifies the child component. The w superscript represents the world frame defined by the center of the main body of the primary satellite. The first term of equation (9) is the same as the reprojection error in equation (5), however the rotation and translations are now dependent on the higher components in the kinematic chain. The last term is also similar to the smoothness constraint in equation (5), however due to the articulation constraints the articulation angle is the only temporally dependent variable.

$$\min_{R,J,\Omega} \sum_{j=1}^{N-1} \left[\underbrace{\sum_{i=1}^{CF} \left[\|W_i - P_i(R_i^{w,c}\Omega^c + T^{w,c})\|^2 \right]}_{\text{Reprojection Error}} + \underbrace{\lambda_{jp} \sum_{k=1}^{12} ln(1+e^{d_k \eta})}_{\text{Joint Penalty}} + \underbrace{\lambda_{sc} \sum_{i=2}^{CF} (1 - cos(\phi_i - \phi_{i-1}))^2}_{\text{Smoothness Constraint}} \right] \tag{9}$$

The rotation matrices and translations are calculated based on the hierarchy of the kinematic chain and the articulation constraints. Define $X = [x_1, x_2, \cdots, x_l]$ as a single chain starting at the root and ending with a component that has no children where each element of X identifies a component and l is the length of the chain. The rotations and translations can be written as follows:

$$R_i^{w,x_k} = \begin{cases} I & , k = 1 \\ R_i^{w,x_{k-1}} R_i^{x_{k-1},x_k} & , \text{otherwise} \end{cases} \tag{10}$$

$$T_i^{w,x_k} = \begin{cases} \mathbf{0} & , k = 1 \\ R_i^{w,x_{k-1}} J^p + T_i^{w,x_{k-1}} - R_i^{w,x_k} J^c & , \text{otherwise} \end{cases} \tag{11}$$

J^p and J^c are the location of the joint in the parent's body frame and the location of the joint in the child's body frame respectively for the joint between x_k and x_{k-1}. Parameterizing using the joint locations forces linkage between the components and reduces the number of optimization variables.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

The rotation between linked frames $R_i^{x_{k-1},x_k}$ is parameterized by two Euler axes, one for the parent (x_{k-1}) and one for the child (x_k), and an Euler angle ϕ_i. Since body frames are chosen arbitrarily during rigid body optimization of each component, there is no reason the articulation axis would be the same between the body frame of the parent and the body frame of the child. This means the frames must be aligned before the articulation angle is applied. An intermediate frame (aa) is used to align the frames and then a rotation matrix is formed using the articulation axis of the parent ($\hat{a}_p^{x_k}$) and the articulation angle (ϕ_i).

$$\begin{aligned} R_i^{x_{k-1},x_k} &= R_i^{x_{k-1},aa} R^{aa,x_k} \\ R^{aa,x_k} &= cos(\psi_a) I + (1 - cos(\psi_a)) \hat{a}_a (\hat{a}_a)^T - sin(\psi_a) \hat{a}_a^{\times} \\ \vec{a}_a &= \hat{a}_{x_{k-1}} \times \hat{a}_{x_k} \\ \hat{a}_a &= \frac{\vec{a}_a}{\|\vec{a}_a\|} \\ \psi_a &= acos(\hat{a}_{x_{k-1}} \cdot \hat{a}_{x_k}) \end{aligned} \tag{12}$$

$$R_i^{x_{k-1},aa} = cos(\phi_i) I + (1 - cos(\phi_i)) \hat{a}_{x_{k-1}} (\hat{a}_{x_{k-1}})^T - sin(\phi_i) \hat{a}_{x_{k-1}}^{\times} \tag{13}$$

The joint penalty in equation (9) encourages the joints to remain close to the point cloud. For each component, a bounding box (BB) containing the points in Ω^n is created and scaled by a factor of 1.5 (γ_{BB}). A soft plus function uses the difference between joint location and the bounding box (d) and a multiplier (η) to penalize the joints (J_p and J_c) if they are outside of the bounding box.

To initialize the optimization for each joint, the information from the rigid body optimization is used. Since each joint is revolute it consists of rotation about a constant axis. A rotation matrix relating the body frame of each component to the world frame is available from rigid body optimization. To find the articulation parameters ($\hat{a}_p$, $\hat{a}_c$, ϕ_i) that most closely match the calculated rotation between components ($R_i^{p,c}$) over common frames, the following cost function is optimized:

$$\min_{M,\phi} \sum_{i=1}^{CF} acos(0.5 trace(R_i^{p,c} (M_i^{p,c})^T) - 1) + \lambda_a \sum_{i=2}^{CF-1} (\phi_{i+1} - 2\phi_i + \phi_{i+1})^2 \tag{14}$$

The first term enforces the single axis of rotation constraint while the second term encourages smooth motion by only penalizing articulation that is not at a constant rate. $R_i^{p,c}$ is calculated from $R_i^{w,p}$ and $R_i^{w,c}$. $M_i^{p,c}$ is calculated from the optimization variables ($\hat{a}_p$, $\hat{a}_c$, ϕ_i) by first aligning the axes and then rotating by ϕ_i as outlined in equation set (12).

As expressed in Yucer et al. [29], the joint location translated into the world frame will coincide for two linked parts.

$$R_i^p J^p + T_i^p = R_i^c J^c + T_i^c, \; \forall i \in CF$$

For components linked by a universal joint there is a single J_p and J_c that best satisfy this constraint. For a revolute joint, any point on the axis will satisfy this constraint. To narrow in on the joint locations that are closest to each other, the norm of the joint locations is also minimized giving the following cost function.

$$\min_{J_p,J_c} \sum_{i=1}^{CF} \|(R_i^p J^p + T_i^p) - (R_i^c J^c + T_i^c)\|^2 + \lambda_b \left[\|J_p\| + \|J_c\|\right] \tag{15}$$

As the kinematic chain grows, the complexity of the problem increases, as does the number of variables. The number of optimization variables in equation (9) is $((12 + CF) \times (N - 1)) + (3 \times P)$. Each articulation parameter ($\hat{a}_p$, $\hat{a}_c$, ϕ, J_p, J_c) for each joint has an influence on each joint below it in the kinematic chain. This means the optimization variables are highly interdependent resulting in many local minima for equation (9). However, the articulation parameters do not have an influence on the components above the joint in the kinematic chain. Using this relationship, a method termed Incremental Joint Addition (IJA) was developed (see Alg. 1). IJA optimizes for the articulation parameters in order, according to the kinematic chain by optimizing for each joint in order and using the results to initialize the next step. Equation (9) is optimized

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

N-1 times, each time another joint is added to the chain. This decreases the complexity since each time a set of parameters is introduced into the optimization they only influence the reprojection error for one component. It also provides a method of further combining components that are likely to be the same rigid body. If the variance of the Euler angle between components is low, they are tested and possilby merged. The details of the IJA implementation are shown in Alg. 1. Results using the IJA method as well as optimizing over all joint parameters together (AJP) are presented in section 3.1.

2.8 Evaluating Results

Algorithm 1 Incremental Joint Addition

1: **procedure** IJA
2: H is table of joints from equation (8)
3: $\mathbf{\Omega} = \{\Omega^1, \Omega^2, \cdots, \Omega^N\}$
4: $f(x)$ =Value of equation (9) at x
5: **while** i $\leq$ M **do**
6: $p = H(i,1)$
7: $c = H(i,2)$
8: $x_0 = \left[x_0, \hat{a}^i_{p_0}, \hat{a}^i_{c_0}, \vec{\phi}^i_0, J^i_{p_0}, J^i_{c_0}\right]$
9: Optimize equation (9) for x using constant shape $\mathbf{\Omega}$
10: $x^s_0 = [x, \Omega^c]$
11: Optimize equation (9) for x^s using constant shape for all except component c
12: **if** $var(\vec{\phi}^i) > \gamma_a$ **then**
13: $f(x)_{org} = f(x)\gamma_m$
14: Combine Ω^c and $\Omega^p \rightarrow \Omega^{pc}$
15: $i = i - 1$
16: Run lines 6-11 with Ω^{pc} as c
17: **if** $f(x) < f(x)_{org}(\gamma_m)$ **then**
18: $\Omega_c = \Omega_{pc}$
19: $H(i,2) = H(i+1,2)$
20: Delete $H(i+1,:)$
21: **end if**
22: **end if**
23: Split $x^s = \left[x, \Omega^{c^*}\right]$
24: $x_0 = x$, $\Omega^c = \Omega^{c^*}$
25: $i = i + 1$
26: **end while**
27: **end procedure**

Multiple metrics are used to evaluate the results. Because the trajectory matrix is simulated from a known point cloud, the position of each point at each time step is known. This allows calculation of the normalized reconstruction error [5] (E_{3D}) which is direct comparison of the calculated point cloud to the known point cloud in the world frame at each time step. E_{3D} is calculated using equation (16) where $\mathbf{S}^{truth}_i$ are the true world frame point locations of all points calculated in image frame i and $\mathbf{S}^{calc}_i$ are the calculated world frame point locations in image frame i.

$$E_{3D} = \frac{\sum_{i=1}^{F} \|\mathbf{S}^{truth}_i - \mathbf{S}^{calc}_i\|}{\sum_{i=1}^{F} \|\mathbf{S}^{truth}_i\|} \tag{16}$$

The calculated kinematic chain may contain more or fewer joints than the truth model. Once the calculated joints are matched to the most appropriate truth joint the articulation parameters can be compared. However, the articulation parameters of the calculated model are in a body frame that is unrelated to the body frame used to create the truth model. To alleviate this, the articulation axis ($E_{\hat{a}}$) and joints (E_J) are compared in the world frame (wf) as follows, where N is the number of joints, and F is the number of frames with data. Since the joints are revolute, any location on the axis is correct, therefore for the joints the error is measured as the distance from the joint to the line represented by the true joint location and articulation axis in the world frame.

$$E_{\hat{a}} = \frac{1}{N} \sum_{j=1}^{N} \left[\frac{1}{F_j} \sum_{i=1}^{F_j} acos(|(\hat{a}^j_{truth,wf})^T \hat{a}^j_{calc,wf}|) \right] \tag{17}$$

$$E_J = \frac{1}{N} \sum_{j=1}^{N} \left[\frac{1}{F_j} \sum_{i=1}^{F_j} \|\hat{a}^j_{truth,wf} \times (J^j_{calc,wf} - J^j_{truth,wf})\| \right] \tag{18}$$

The articulation angle is evaluated using the error in the range of angles covered, $E_{\phi_{range}}$, and the rate of change of the angle $E_{\dot{\phi}}$.

$$E_{\phi_{range}} = \frac{1}{N} \sum_{j=1}^{N} |\Delta^j_{truth} - \Delta^j_{calc}| \tag{19}$$

$$\Delta^j = max(\vec{\phi}^j) - min(\vec{\phi}^j) \tag{20}$$

$$E_{\dot{\phi}} = \frac{1}{N} \sum_{j=1}^{N} \left[\frac{1}{F_j} \sum_{i=2}^{F_j} |\dot{\phi}^i_{truth} - \dot{\phi}^i_{calc}| \right] \tag{21}$$

$$\dot{\phi}^i = \phi^{i-1} - \phi^i \tag{22}$$

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

In addition to evaluating the results directly, it may be interesting to understand how the satellite is capable of moving, or more generally, what area around the satellite are the appendages capable of reaching. This can be termed the satellite's workspace. To calculate the satellite's workspace, the space around the satellite is discretized into a grid of $m \times m \times m = M$ points. Each joint is moved through its range of articulation angles at Y increments. A convex hull is created encompassing each shape in the world frame in the current increment and the previous increment. The grid points in M that fall within the convex hull [3] are annotated as covered. The collection of points in M that are covered represents the workspace of the satellite. The complexity of this calculation scales exponentially with the number of increments and the length of the longest portion of the kinematic chain K. If the computation time for a single increment of a 2 link chain is Q then Q scales as Y^K. This means a 6 link chain incremented 10 times requires computation time of $10^6 Q$. The computational requirement can be large, but can be managed by using only a handful of increments. Both the calculated workspace and the true workspace are found in this manner. They are compared using two percentages. P_{wc} is the percentage of the truth workspace covered by the calculated model and P_{woc} is the percentage of additional space covered by the calculated model. These values are calculated as follows where B is an $M \times 1$ vector with $B_i = 1$ when grid point i is covered and $B_i = 0$ when grid point i is not covered. $\odot$ denotes element-wise multiplication.

$$P_{wc} = \frac{\sum B^{calc} \odot B^{truth}}{\sum B^{truth}} \tag{23}$$

$$P_{woc} = \frac{\sum B^{calc} \odot |B^{truth} - 1|}{\sum B^{truth}} \tag{24}$$

Note that the workspace calculation does not consider physical limits to motion imposed by other components. In other words, two components may occupy the same physical space during some combination of motions.

3 RESULTS AND LIMITATIONS

3.1 Results

The algorithms performance was measured with various joint activity using a 2×1 NMC [19] in geostationary orbit. The seleted NMC was 2,000 units (in-track) by 1,000 units (radial) when projected into the orbital plane with ± 50 units of cross track and was phased for the best illumination conditions. The results presented were conducted under winter solstice illumination conditions, however trials were also conducted under summer solstice and equinox illumination with similar results. Fig. 1b shows a schematic of the simulated satellite and its articulating joints. In all cases, the panels articulate linearly from -1.3 radians to 1.3 radians. Joints in the appendage that are active articulate linearly from -0.94 radians to 0.94 radians.

Table 1: Results for Single NMC at Winter Solstice.

Input/Truth			Results									
AP method	# of Comp.	# of Comp. in appendage	# of Comp.	# of Comp. in appendage	E_{3D} (%)	P_{wc} (%)	P_{woc} (%)	$E_{\hat{a}}$ (rad.)	E_J (units)	$E_{\Delta\phi}$ (rad.)	$E_{\dot{\phi}}$ (rad. per frame)	Time for AP Opt (s)
AJP	3	1	8	3	0.29%	80%	9.4%	0.00066	0.0015	0.0010	0.000063	678
IJA	3	1	6	1	0.30%	94%	8.2%	0.0037	0.0044	0.00027	0.000049	6,213
AJP	4	2	10	5	0.72%	97%	8.9%	0.029	0.059	0.023	0.0011	646
IJA	4	2	7	2	0.31%	97%	0.82%	0.0046	0.0044	0.0066	0.000098	7,287
AJP	5	3	10	4	0.21%	96%	0.87%	0.0084	0.013	0.041	0.00048	969
IJA	5	3	8	3	0.25%	97%	0.9%	0.0057	0.014	0.043	0.00052	1,163
AJP	6	4	11	4	0.39%	97%	1.4%	0.013	0.024	0.051	0.00077	1,280
IJA	6	4	9	4	0.32%	97%	0.87%	0.0099	0.016	0.040	0.00068	1,000
AJP	7	5	13	6	0.81%	92%	1.6%	0.043	0.046	0.075	0.0015	1,881
IJA	7	5	11	5	0.89%	93%	1.5%	0.062	0.047	0.063	0.0014	2,466
AJP	8	6	15	9	1.2%	63%	5.4%	0.15	0.051	0.16	0.0027	4,427
IJA	8	6	11	6	0.82%	93%	0.70%	0.074	0.031	0.14	0.0016	3,949

The AJP and IJA methods of optimizing for articulation parameters were evaluated. AJP results and IJA results are comparable for most metrics and for most trials as shown in Table 1. While there are cases

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

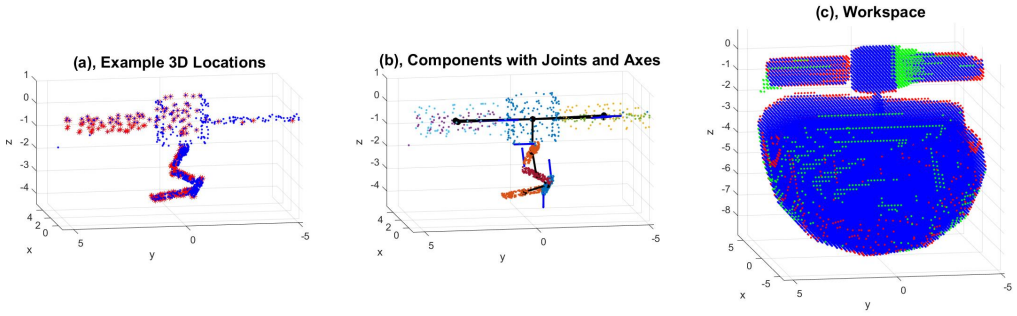

Fig. 5. Example results. a) Truth (blue ·) and calculated (red ∗) 3D position of points for an example frame. b) Calculated points, kinematic chain, joint locations, and axes for an example frame. Points in each component are a different color. c) Grid positions in both true and calculated workspace (blue ·), Grid positions in only true workspace (red ·), Grid positions in only calculated workspace (green ·).

when each performs better, there are no instances when AJP performs significantly better than IJA. There are instances, such as when all 6 joints are active, when the IJA performs significantly better than IJA. The ability of IJA to merge components contributes to its improved performance and allows the number of components to be closer to truth than AJP in nearly all cases. These advantages may be worth the added computational cost, particularly as the number of joints increases.

The articulation parameter metrics ($E_{\hat{a}}$, E_J, $E_{\Delta\phi}$, and $E_{\dot{\phi}}$) shown in Table 1 are calculated for only the joints on the appendage. The algorithm is able to characterize the articulation parameters when all 6 joints are active, however the average error per joint increases as the length of the kinematic chain increases. Detailed plots of each trial are not presented, however the results with four active joints on the articulating appendage are presented in Fig. 5.

A fly-by inspection trajectory was also analyzed in which the inspector satellite passed the primary satellite offset by 500 units. Simulated imagery begins and ends when the satellite is 1000 units from the primary satellite. In the example shown in Fig. 6 the calculated panel motion is about an axis perpendicular to its true articulation axis and parallel to the axis about which the camera rotates. The reason for these errors stems back to the ray space optimization used to initialize the shape. Since equation (2) minimizes the distance between temporally adjacent points, when the motion of the camera is slower than the articulated motion the optimization may find a minimum closer to the camera's motion than the the articulated motion. Since the ray space results are used for initialization of the rigid body optimization, this can result in a rigid body optimization solution that is closer to the camera motion. The rigid body optimization result are then used in initialization of articulation parameter optimization causing the final results to fall into a local minimum. In the case shown in Fig. 6, the motion of the joints in the appendage (1.9 radians over 100 frames) are slower than the motion of the panel joints (2.5 radians over 100 frames). The rotation of the camera is 2.1 radians over 100 frames. This is faster than the appendage motion but slower than the panel motion allowing the appendage to be initiated/calculated correctly ($E_{3D} = 1.7\%$) while the panel has the large errors evident in Fig. 6. A potential solution to this issue is to use another method to initialize the shape such an estimate assuming a scaled orthographic camera model. This will be investigated in future work.

Table 2: Computational Time.

Section	Approximate Computational Time
Simulate Data	16 sec
Ray space optimization	3.5 min
Segmentation	1 min
Rigid body optimization	31 min
Combine Segments	4.3 hrs
AP optimization (AJP)	6 min
AP optimization (IJA)	35 min
Total	5-5.5 hrs

Table 2 shows the approximate computational time for this method, using a simulation with 100 frames and 1,100 points with 7 active joints for a total of 8 components.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

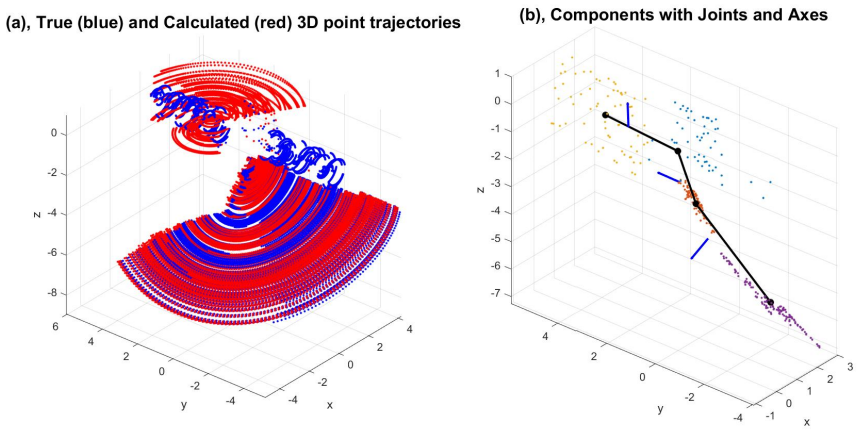

Fig. 6. Results from a fly-by inspection trajectory. a) True (blue ·) and calculated (red ·) 3D point trajectories. b) 3D points a particular time frame with articulation axes shown at the joint position. The axis for the panel (top left) should be parallel to the y-axis.

3.2 Limitations and Future Work

This method is capable of charaterizing satellite articulation in the scenarios tested, however limitations exist.

- There are a number of parameters that require adjustment based on the specifics of the problem. These parameters are shown in Table 3 along with the method of determining an appropriate setting. Noise analysis (NA) means that the best parameter setting is based on the amount of noise in the trajectory matrix and the inspector satellite route. Prior problem knowledge (PPK) means that some information regarding the particular problem, such as an estimate on the number of components, should be used to select an appropriate value. Cost function ratios (CFR) means that the ratio of the different terms in the cost function should be used to scale the value appropriately. Point density (PD) means the number of points found/tracked should dictate the criteria for removing points. Risk assessment (RA) means the risk in having more or less than the correct number of components should be assessed. Investigating methods of automatically adjusting these parameters is an area for future work.

- The current implementation only calculates parameters for a component when the component is visible and when all components above it in the kinematic chain are visible. This could be highly limiting for certain cases in which an appendage is shadowed or occluded for a significant number of frames. Methods of reducing this limitation is an area for future work.

- All joints are assumed to be revolute. Errors will occur if a joint is actually prismatic or universal. They will also likely occur if a component in the chain is not visible. Even if the method effectively skips the missing link there is no reason for the demonstrated motion of the next component to adhere to the constraints of a revolute joint connection to the previous component.

- This method is not robust to kinematic chains which contain cycles.

- As currently implemented, this method requires the main body of the satellite to be motionless in the world frame. While this is certainly a limitation, it is likely this limitation could be removed within the same framework. This will be investigated in future work.

- It is assumed that the motion of the inspector satellite is known. While this is likely to be the case generally, the exact relative position may not be known. The robustness of this method to noise in the trajectory matrix or the inspector satellite route has not been investigated.

- Real imagery and the challenges of feature point recognition and tracking in space illumination conditions have not been investigated in this work.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Table 3: Variable parameters. Method of determination: NA = Noise analysis; PPK=Prior problem knowledge; CFR=Cost function ratios; PD=Point density; RA=Risk assessment

Parameter	Name/description	Value	Method of determination
γ_s	Stationary threshold: determines when a point is considered stationary	1^{-6}	NA
k	Number of segments: determines the number of segments for spectral clustering. (sec. 2.3)	15 to 30	PPK
λ_{rb}	Rigid body optimization smoothness weight. (egn. (5))	1^{-8}	CFR
γ_f	Reprojection error fair share multiplier for outlier rejection. (sec. 2.4)	20	PD
γ_r	Range rejection multiplier for outlier rejection. (sec. 2.4)	4	PD, PPK
γ_c	Segment merge criteria: allowable increase in function value for acceptable merge. (sec. 2.5)	1.5	RA
γ_{BB}	Joint bounding box expansion: determines the size of the bounding box around the component shape. (sec. 2.7)	0.5	PPK
η	Coefficient in joint penalty. (eqn. (9))	100	CFR
λ_{jp}	Articulation parameter optimization joint location penalty weight. (eqn. (9))	1^{-4}	CFR
λ_{sc}	Articulation parameter optimization smoothness weight. (eqn. (9))	0.5	CFR
λ_a	Rotation parameter initialization smoothness weight. (eqn. (14))	5	CFR
λ_b	Joint location initialization closeness weight. (eqn. (15))	5	CFR
γ_a	Minimum articulation angle variance (IJA method only). (alg. 1)	0.05 rad	RA
γ_m	Segment merge criteria: allowable increase in function value for acceptable merge (IJA method only). (alg. 1)	1.1	RA

- The trajectory matrix assumes points are reacquired and identified as the same point after they drop out of the frame. This is unrealistic, particularly if the point drops out for a significant number of frames. This limitation will be removed in in future work.

4 CONCLUSION

This work provides a method of detecting and characterizing the articulated motion of a satellite from an inspector satellite in close proximity. Assuming the inspector satellite's trajectory is known, and using concepts presented in Yucer et. al.[29] a method is developed which relies on multiple static optimization routines. First, ray space optimization (equation (2)) is used to give an approximation of the 3D location of points in the world frame. Next, points are segmented and rigid body optimization (equation (5)) is used to approximate the rigid body motion of each segment. Inspector routes such as NMCs produce trajectory matrices with a high percentage of missing data causing over segmentation. To combat this, a routine is run to combine segments that are likely part of the same rigid body. Once all appropriate segments are combined, the kinematic chain is built and revolute articulation constraints are applied through equations (10) and (11). Next, two methods are investigated for applying articulation constraints and optimizing for articulation parameters using equation (9).

Results demonstrate the success of this method in characterizing the articulation of a nominal satellite with an appendage containing various active joints. The method is capable of building a model of the primary satellite containing the correct number of components in the appendage and containing the articulation axes, angles, and joint locations without prior knowledge of the primary satellite using only feature point locations and the inspection route. Using the presented metrics, this method can be evaluated against other methods for characterizing articulation. In development of this method the importance of analytical derivatives for optimization routines with many variables is demonstrated through a 99% reduction in computational time compared to finite differences. In addition, the resulting model can be represented with a fraction of the memory required by the full imagery or even the trajectory matrix which could be beneficial if communication bandwidth with the ground is limited. Future work will focus on minimizing the limitations presented in the previous section and expanding testing to additional inspection routes and illumination conditions.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

ACKNOWLEDGMENTS

The authors would like to acknowledge Dr Frank Chavez and others at the Air Force Research Laboratory Space Vehicles directorate for their guidance in this research.

APPENDIX A

Derivation for the derivatives used in rigid body optimization (equation (5)) are as follows:

$$f(x) = \sum_{i=1}^{F} \|W_i - P_i(R_i\mathbf{\Omega} + \tilde{T}_i)\|^2 + \lambda_{rb}\sum_{i=2}^{F}\left[acos(.5(trace(R_iR_{i-1}^T) - 1)) + \|T_i - T_{i-1}\|^2\right] \quad (25)$$

$$x = [\mathbf{a}, \phi, \mathbf{T}, \mathbf{\Omega}]^T \quad (26)$$

The optimization variables are contained in the vector x which consist of a $3F \times 1$ vector of Euler axes (not normalized) $\mathbf{a} = [\vec{a}_1^T, \vec{a}_2^T, \cdots, \vec{a}_F^T]$, a $F \times 1$ vector of Euler angles $\phi = [\phi_1, \phi_2, \cdots, \phi_F]$, a $3F \times 1$ vector of translations $\mathbf{T} = [\vec{T}_1^T, \vec{T}_2^T, \cdots, \vec{T}_F^T]$, and a $3P \times 1$ vector of shape coordinates $\mathbf{\Omega} = [\vec{s}_1^T, \vec{s}_2^T, \cdots, \vec{s}_P^T]$. These optimization variables translate to the variables in the cost function as follows.

$$R_i = cos(\phi_i)\mathbf{I} + (1 - cos(\phi_i))\hat{a}_i\hat{a}_i^T - sin(\phi_i)\hat{a}_i^x \quad (27)$$

$$\hat{a}_i = \frac{\vec{a}_i}{\|\vec{a}_i\|} \quad (28)$$

$$\tilde{T}_i = T_i - R_i\bar{\Omega} \quad (29)$$

$$\bar{\Omega} = \frac{1}{P}\sum_{j=1}^{P}\vec{s}_j \quad (30)$$

The gradient of the function consists of the partial derivatives of each of the optimization variables $\frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial \mathbf{a}}, \frac{\partial f}{\partial \phi}, \frac{\partial f}{\partial \mathbf{T}}, \frac{\partial f}{\partial \mathbf{\Omega}}\right]^T$. The chain rule is used to relate each of these to the components of the cost function. Start with the partial between the rotation matrix R_i and the rotation parameters ($\vec{a}_i$ and ϕ_i).

$$\frac{\partial f}{\partial \vec{a}_i} = \frac{\partial f}{\partial R_i}\frac{\partial R_i}{\partial \hat{a}_i}\frac{\partial \hat{a}_i}{\partial \vec{a}_i} \quad (31)$$

$$\hat{a}_i = [\hat{a}_i^{(1)}, \hat{a}_i^{(2)}, \hat{a}_i^{(3)}]^T \quad (32)$$

$$R_i = \begin{bmatrix} R_i^{(1)} & R_i^{(2)} & R_i^{(3)} \\ R_i^{(4)} & R_i^{(5)} & R_i^{(6)} \\ R_i^{(7)} & R_i^{(8)} & R_i^{(9)} \end{bmatrix} \quad (33)$$

$$\frac{\partial R_i}{\partial \hat{a}_i} = \begin{bmatrix} \frac{\partial R_i^{(1)}}{\partial \hat{a}_i^{(1)}} & \frac{\partial R_i^{(1)}}{\partial \hat{a}_i^{(2)}} & \frac{\partial R_i^{(1)}}{\partial \hat{a}_i^{(3)}} \\ \frac{\partial R_i^{(2)}}{\partial \hat{a}_i^{(1)}} & \frac{\partial R_i^{(2)}}{\partial \hat{a}_i^{(2)}} & \frac{\partial R_i^{(2)}}{\partial \hat{a}_i^{(3)}} \\ \vdots & \vdots & \vdots \\ \frac{\partial R_i^{(9)}}{\partial \hat{a}_i^{(1)}} & \frac{\partial R_i^{(9)}}{\partial \hat{a}_i^{(2)}} & \frac{\partial R_i^{(9)}}{\partial \hat{a}_i^{(3)}} \end{bmatrix} \quad (34)$$

(35)

The derivatives of R_i with respect to $\hat{a}_i$ are as follows. They are written as 3×3 matrices, however they

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

should be reshaped to 9×3 to be used in the chain rule as written.

$$\frac{\partial R_i}{\partial \hat{a}_i^{(1)}} = \begin{bmatrix} 2(1-cos(\phi_i))\hat{a}_i^{(1)} & (1-cos(\phi_i))\hat{a}_i^{(2)} & (1-cos(\phi_i))\hat{a}_i^{(3)} \\ (1-cos(\phi_i))\hat{a}_i^{(2)} & 0 & sin(\phi_i) \\ (1-cos(\phi_i))\hat{a}_i^{(3)} & -sin(\phi_i) & 0 \end{bmatrix} \tag{36}$$

$$\frac{\partial R_i}{\partial \hat{a}_i^{(2)}} = \begin{bmatrix} 0 & (1-cos(\phi_i))\hat{a}_i^{(1)} & -sin(\phi_i) \\ (1-cos(\phi_i))\hat{a}_i^{(2)} & 2(1-cos(\phi_i))\hat{a}_i^{(2)} & (1-cos(\phi_i))\hat{a}_i^{(2)} \\ sin(\phi_i) & (1-cos(\phi_i))\hat{a}_i^{(3)} & 0 \end{bmatrix} \tag{37}$$

$$\frac{\partial R_i}{\partial \hat{a}_i^{(3)}} = \begin{bmatrix} 0 & sin(\phi_i) & (1-cos(\phi_i))\hat{a}_i^{(1)} \\ -sin(\phi_i) & 0 & (1-cos(\phi_i))\hat{a}_i^{(2)} \\ (1-cos(\phi_i))\hat{a}_i^{(1)} & (1-cos(\phi_i))\hat{a}_i^{(2)} & 2(1-cos(\phi_i))\hat{a}_i^{(3)} \end{bmatrix} \tag{38}$$

The Euler axis should be constrained to be unit length. Instead of enforcing unit length as a hard constraint in the optimization, the vector is normalized before use in the cost function. The following derivatives relate the Euler axis $\hat{a}_i$ to the optimization variable $\vec{a}_i$. I represents the 3×3 identity matrix.

$$\frac{\partial \hat{a}_i}{\partial \vec{a}_i} = \begin{bmatrix} \frac{\partial \hat{a}_i^{(1)}}{\partial a_i^{(1)}} & \frac{\partial \hat{a}_i^{(1)}}{\partial a_i^{(2)}} & \frac{\partial \hat{a}_i^{(1)}}{\partial a_i^{(3)}} \\ \frac{\partial \hat{a}_i^{(2)}}{\partial a_i^{(1)}} & \frac{\partial \hat{a}_i^{(2)}}{\partial a_i^{2}} & \frac{\partial \hat{a}_i^{(2)}}{\partial a_i^{(3)}} \\ \frac{\partial \hat{a}_i^{(3)}}{\partial a_i^{(3)}} & \frac{\partial \hat{a}_i^{(3)}}{\partial a_i^{(2)}} & \frac{\partial \hat{a}_i^{(3)}}{\partial a_i^{(3)}} \end{bmatrix} \tag{39}$$

$$\frac{\partial \hat{a}_i}{\partial \vec{a}_i} = \frac{1}{\|\hat{a}_i\|} I - \frac{1}{\|\hat{a}_i\|^3}(\hat{a}_i \hat{a}_i^T) \tag{40}$$

The derivative of R_i with respect to the Euler angle ϕ_i is as follows.

$$\frac{\partial R_i}{\partial \phi_i} = \begin{bmatrix} -sin(\phi_i) & cos(\phi_i)\hat{a}_i^{(3)} & -cos(\phi_i)\hat{a}_i^{(2)} \\ -cos(\phi_i)\hat{a}_i^{(3)} & -sin(\phi_i) & -cos(\phi_i)\hat{a}_i^{(1)} \\ cos(\phi_i)\hat{a}_i^{(2)} & -cos(\phi_i)\hat{a}_i^{(1)} & -sin(\phi_i) \end{bmatrix} + sin(\phi_i)(\hat{a}_i \hat{a}_i^T) \tag{41}$$

The partial derivative of the centered translation $\tilde{T}_i$ with respect to the rotation matrix R_i, the actual translation T_i, and the shape $\mathbf{\Omega}$ must also be calculated. The symbol $\otimes$ denotes the Kronecker product.

$$\frac{\partial \tilde{T}_i}{\partial T_i} = I \tag{42}$$

$$\frac{\partial \tilde{T}_i}{\partial R_i} = -I \otimes (\bar{\Omega})^T \tag{43}$$

$$\frac{\partial \tilde{T}_i}{\partial \Omega_j} = -\frac{1}{P} R_i \tag{44}$$

Now that the rotation matrix has been related to the optimization variables, it must now be related to the cost function. The cost function can be broken into two parts: D is the reprojection error, and L is the smoothness constraint. Note that $\|\cdot\|$ is the Frobenius norm, or the summation of each element squared.

$$D = \sum_{i=1}^{F} \|W_i - P_i(R_i \mathbf{\Omega} + \tilde{T}_i)\|^2 \tag{45}$$

$$L = \lambda_{rb} \sum_{i=2}^{F} acos(.5(trace(R_i R_{i-1}^T) - 1)) + \|T_i - T_{i-1}\|^2 \tag{46}$$

$$\frac{\partial f}{\partial R} = \frac{\partial D}{\partial R} + \frac{\partial L}{\partial R} \tag{47}$$

$$\frac{\partial f}{\partial T} = \frac{\partial D}{\partial T} + \frac{\partial L}{\partial T} \tag{48}$$

$$\frac{\partial f}{\partial \Omega} = \frac{\partial D}{\partial \Omega} \tag{49}$$

The reprojection error for frame i is D_i which is the summation of the reprojection error for the P points available in frame i. The terms u and v are the image coordinates in the horizontal and vertical dimensions

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

respectively.

$$D_i = \sum_{u,v} \sum_{j=1}^{P} G_{i,j}^2 \tag{50}$$

$$G_{i,j} = \begin{bmatrix} u_j \\ v_j \end{bmatrix} - P_i(R_i \Omega_j + \tilde{T}_i) \tag{51}$$

$$\frac{\partial D_i}{\partial R_i} = \sum_{u,v} \sum_{j=1}^{P} 2 G_{i,j} \left[\frac{\partial G_{i,j}}{\partial R_i} + \frac{\partial G_{i,j}}{\partial \tilde{T}_i} \frac{\partial \tilde{T}_i}{\partial R_i} \right] \tag{52}$$

Define $\tilde{G}$ as the second term of G written in homogeneous coordinates. f is the focal length of the camera using a pinhole camera model.

$$\tilde{G}_{i,j} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \left[R_{CW} | \mathbf{t} \right] \begin{bmatrix} R_i & T_i \\ 0,\ 0,\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_j \\ 1 \end{bmatrix} \tag{53}$$

$$\tilde{G}_{i,j} = \begin{bmatrix} \tilde{G}1_{i,j} \\ \tilde{G}2_{i,j} \\ \omega \end{bmatrix} \tag{54}$$

$$G_{i,j} = \begin{bmatrix} u_j \\ v_j \end{bmatrix} - \begin{bmatrix} \frac{\tilde{G}1_{i,j}}{\omega} \\ \frac{\tilde{G}2_{i,j}}{\omega} \end{bmatrix} \tag{55}$$

The rotation, translation and shape derivatives are then calculated as follows using Matlab matrix notation where appropriate. The symbol $\odot$ denotes element-wise multiplication.

Rotation derivatives:

$$\frac{\partial G_{i,j}}{\partial R_i} = - \begin{bmatrix} \frac{\omega \frac{\partial \tilde{G}1_{i,j}}{\partial R_i} - \tilde{G}1_{i,j} \frac{\partial \omega}{\partial R_i}}{\omega^2} \\ \frac{\omega \frac{\partial \tilde{G}2_{i,j}}{\partial R_i} - \tilde{G}2_{i,j} \frac{\partial \omega}{\partial R_i}}{\omega^2} \end{bmatrix} \tag{56}$$

$$\frac{\partial \tilde{G}1_{i,j}}{\partial R_i} = - \begin{bmatrix} \frac{\partial \tilde{G}1_{i,j}}{\partial R_i^1}, & \frac{\partial \tilde{G}1_{i,j}}{\partial R_i^2}, & \cdots & \frac{\partial \tilde{G}1_{i,j}}{\partial R_i^9} \end{bmatrix} \tag{57}$$

$$\frac{\partial \tilde{G}1_{i,j}}{\partial R_i} = -f \left[(R_{CW}^{(1,:)} \otimes [1,1,1]) \odot (s_j^T \otimes [1,1,1]) \right] \tag{58}$$

$$\frac{\partial \tilde{G}2_{i,j}}{\partial R_i} = -f \left[(R_{CW}^{(2,:)} \otimes [1,1,1]) \odot (s_j^T \otimes [1,1,1]) \right] \tag{59}$$

$$\frac{\partial \tilde{\omega}_{i,j}}{\partial R_i} = - \left[(R_{CW}^{(3,:)} \otimes [1,1,1]) \odot (s_j^T \otimes [1,1,1]) \right] \tag{60}$$

Translation derivatives:

$$\frac{\partial D_i}{\partial T_i} = \sum_{u,v} \sum_{j=1}^{P} 2 G_{i,j} \frac{\partial G_{i,j}}{\partial \tilde{T}_i} \frac{\partial \tilde{T}_i}{\partial T_i} \tag{61}$$

$$\frac{\partial G_{i,j}}{\partial \tilde{T}_i} = - \begin{bmatrix} \frac{\omega \frac{\partial \tilde{G}1_{i,j}}{\partial \tilde{T}_i} - \tilde{G}1_{i,j} \frac{\partial \omega}{\partial \tilde{T}_i}}{\omega^2} \\ \frac{\omega \frac{\partial \tilde{G}2_{i,j}}{\partial \tilde{T}_i} - \tilde{G}2_{i,j} \frac{\partial \omega}{\partial \tilde{T}_i}}{\omega^2} \end{bmatrix} \tag{62}$$

$$\frac{\partial \tilde{G}1_{i,j}}{\partial \tilde{T}_i} = -f R_{CW}^{(1,:)} \tag{63}$$

$$\frac{\partial \tilde{G}2_{i,j}}{\partial \tilde{T}_i} = -f R_{CW}^{(2,:)} \tag{64}$$

$$\frac{\partial \omega}{\partial \tilde{T}_i} = -R_{CW}^{(3,:)} \tag{65}$$

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Shape derivatives:

$$\frac{\partial D}{\partial \Omega_j} = \sum_{u,v}\sum_{i=1}^{F} 2G_{i,j}\left[\frac{\partial G_{i,j}}{\partial \Omega_j} + \frac{\partial G_{i,j}}{\partial \tilde{T}_i}\frac{\partial \tilde{T}_i}{\partial \Omega_j}\right] \tag{66}$$

$$\frac{\partial G_{i,j}}{\partial \Omega_j} = -\begin{bmatrix} \frac{\omega \frac{\partial \tilde{G1}_{i,j}}{\partial \Omega_j} - \tilde{G1}_{i,j}\frac{\partial \omega}{\partial \Omega_j}}{\omega^2} \\ \frac{\omega \frac{\partial \tilde{G2}_{i,j}}{\partial \Omega_j} - \tilde{G2}_{i,j}\frac{\partial \omega}{\partial \Omega_j}}{\omega^2} \end{bmatrix} \tag{67}$$

$$\frac{\partial \tilde{G1}_{i,j}}{\partial \Omega_j} = -fR_{CW}^{(1,:)}R_i \tag{68}$$

$$\frac{\partial \tilde{G2}_{i,j}}{\partial \Omega_j} = -fR_{CW}^{(2,:)}R_i \tag{69}$$

$$\frac{\partial \omega}{\partial \Omega_j} = -R_{CW}^{(3,:)}R_i \tag{70}$$

The smoothness constraint is only applied to adjacent frames, so the derivatives are different if frame i is the first frame in a sequence, the middle frame in a sequence or the last frame in a sequence. For the first frame in a sequence,

$$L_i = \lambda_{rb}\left[acos(.5(trace(R_{i+1}R_i^T)-1)) + \|T_{i+1}-T_i\|\right] \tag{71}$$

$$U_i = 0.5(trace(R_iR_{i-1}^T)-1) \tag{72}$$

$$\frac{\partial L_i}{\partial R_i} = -\frac{\lambda_{rb}}{\sqrt{1-U_{i+1}^2}}(.5R_{i+1}) \tag{73}$$

$$V_i = \|T_i - T_{i-1}\| \tag{74}$$

$$= \sqrt{(T_i^1-T_{i-1}^1)^2 + (T_i^2-T_{i-1}^2)^2 + (T_i^3-T_{i-1}^3)^2} \tag{75}$$

$$\frac{\partial L_i}{\partial T_i} = -\frac{\lambda_{rb}}{2\sqrt{V_{i+1}^2}}\left[-2(T_{i+1})^T + 2(T_i)^T\right] \tag{76}$$

For middle frames in a sequence,

$$L_i = \lambda_{rb}\left[acos(.5(trc(R_{i+1}R_i^T)-1)) + \|T_{i+1}-T_i\| + acos(.5(trc(R_iR_{i-1}^T)-1)) + \|T_i-T_{i-1}\|\right] \tag{77}$$

$$\frac{\partial L_i}{\partial R_i} = -\frac{\lambda_{rb}}{\sqrt{1-U_i^2}}(.5R_{i-1}) - \frac{\lambda_{rb}}{\sqrt{1-U_{i+1}^2}}(.5R_{i+1}) \tag{78}$$

$$\frac{\partial L_i}{\partial T_i} = -\frac{\lambda_{rb}}{2\sqrt{V_i^2}}\left[-2(T_i)^T + 2(T_{i-1})^T\right] - \frac{\lambda_{rb}}{2\sqrt{V_{i+1}^2}}\left[-2(T_{i+1})^T + 2(T_i)^T\right] \tag{79}$$

For last frames in a sequence,

$$L_i = \lambda_{rb}\left[acos(.5(trc(R_iR_{i-1}^T)-1)) + \|T_i-T_{i-1}\|\right] \tag{80}$$

$$\frac{\partial L_i}{\partial R_i} = -\frac{\lambda}{\sqrt{1-U_i^2}}(.5R_{i-1}) \tag{81}$$

$$\frac{\partial L_i}{\partial T_i} = -\frac{\lambda}{2\sqrt{V_i^2}}\left[-2(T_i)^T + 2(T_{i-1})^T\right] \tag{82}$$

APPENDIX B

Derivation for the derivatives used in articulation parameter optimization (equation (9)) are as follows:

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

$$f(x) = \sum_{j=1}^{N-1} \left[\underbrace{\sum_{i=1}^{CF} \left[\|W_i - P_i(R_i^{w,c_j}\Omega^{c_j} + T_i^{w,c_j})\|^2 \right]}_{\text{Reprojection Error, A}} + \underbrace{\lambda_{jp} \sum_{k=1}^{12} ln(1 + e^{d_k \eta})}_{\text{Joint Penalty, B}} + \underbrace{\lambda_{sc} \sum_{i=2}^{CF} (1 - cos(\phi_i - \phi_{i-1}))^2}_{\text{Smoothness Constraint, SC}} \right] \quad (83)$$

$$x = [\Omega^{c_1}, \vec{a}^{p_1}, \vec{a}^{c_1}, \phi^1, J^{p_1}, J^{c_1}, \cdots, \Omega^{c_{N-1}}, \vec{a}^{p_{N-1}}, \vec{a}^{c_{N-1}}, \phi^{N-1}, J^{p_{N-1}}, J^{c_{N-1}}]^T \quad (84)$$

Many of the derivatives for the articulation parameter cost function are the same as for rigid body optimization. The primary difference is finding the partial derivatives of the rotation matrix used for each component with respect to the optimization variables lower in the kinematic chain. For each parameter that defines joint j between components x_k and x_{k-1}, the derivatives must be calculated for each component that contains x_{k-1} in its hierarchy. Define X^k as the set of components influenced by component x_k, or in other words X^k is the set of components below component x_k in the kinematic chain.

$$\frac{\partial f}{\partial R_i^{w,x_k}} = \sum_{q=1}^{X^k} \frac{\partial f}{\partial R_i^{w,x_q}} \quad (85)$$

$$\frac{\partial f}{\partial T_i^{w,x_k}} = \sum_{q=1}^{X^k} \frac{\partial f}{\partial T_i^{w,x_q}} \quad (86)$$

To calculate the derivatives of the rotation matrices R_i^{w,x_k} and the translation vectors T_i^{w,x_k} a hierarchical approach is required. Define $X = [x_1, x_2, \cdots, x_l]$ as a single chain starting at the root and ending with a component that has no children where each element of X identifies a component, x_k is the parent and x_{k-1} is the child for any particular joint.

$$\frac{\partial R_i^{w,x_k}}{\partial R_i^{x_{l-1},x_l}} = \begin{cases} \frac{\partial R_i^{w,x_k}}{\partial R_i^{x_{l-1},x_l}} & k = l \\ \frac{\partial R_i^{w,x_k}}{\partial R_i^{w,x_{k-1}}} \frac{\partial R_i^{w,x_{k-1}}}{\partial R_i^{x_{l-1},x_l}} & k > l \end{cases} \quad (87)$$

$$\frac{\partial T_i^{w,x_k}}{\partial R_i^{x_{l-1},x_l}} = \begin{cases} \frac{\partial T_i^{w,x_k}}{\partial R_i^{w,x_k}} & k = l \\ \frac{\partial T_i^{w,x_k}}{\partial R_i^{w,x_{k-1}}} \frac{\partial R_i^{w,x_{k-1}}}{\partial R_i^{x_{l-1},x_l}} + \frac{\partial T_i^{w,x_k}}{\partial T_i^{w,x_{k-1}}} \frac{\partial T_i^{w,x_{k-1}}}{\partial R_i^{x_{l-1},x_l}} + \frac{\partial T_i^{w,x_k}}{\partial R_i^{w,x_k}} & k > l \end{cases} \quad (88)$$

To calculate these partial derivatives, the derivative relating the actual optimization parameters to the rotation matrices must be calculated. The symbol $\otimes$ denotes the Kronecker product.

$$\frac{\partial R_i^{w,x_k}}{\partial R_i^{x_{k-1},x_k}} = R_i^{w,x_{k-1}} \otimes I \quad (89)$$

$$\frac{\partial R_i^{w,x_k}}{\partial R_i^{w,x_{k-1}}} = I \otimes (R_i^{x_{k-1},x_k})^T \quad (90)$$

From equation set (12) the derivatives of $R_i^{x_{k-1},x_k}$ with respect to rotational optimization variables for the joint between x_{k-1} and x_k ($\hat{a}_p^{x_k,x_{k-1}}$, $\hat{a}_c^{x_k,x_{k-1}}$, $\phi_i^{x_k,x_{k-1}}$) can be found.

$$\frac{\partial R_i^{x_{k-1},x_k}}{\partial \hat{a}_p^{x_k,x_{k-1}}} = (R_i^{x_{k-1},aa} \otimes I) \frac{\partial R_i^{aa,x_k}}{\partial \hat{a}_p^{x_k,x_{k-1}}} + (I \otimes (R^{aa,x_k})^T) \frac{\partial R_i^{x_{k-1},aa}}{\partial \hat{a}_p^{x_k,x_{k-1}}} \quad (91)$$

$$\frac{\partial R_i^{aa,x_k}}{\partial \hat{a}_p^{x_k,x_{k-1}}} = \frac{\partial R_i^{aa,x_k}}{\partial \hat{a}_a^{x_k,x_{k-1}}} \frac{\partial \hat{a}_a^{x_k,x_{k-1}}}{\partial \vec{a}_a^{x_k,x_{k-1}}} \frac{\partial \vec{a}_a^{x_k,x_{k-1}}}{\partial \hat{a}_p^{x_k,x_{k-1}}} + \frac{\partial R_i^{aa,x_{k-1}}}{\partial \psi_a^{x_k,x_{k-1}}} \frac{\partial \psi_a^{x_k,x_{k-1}}}{\partial \hat{a}_p^{x_k,x_{k-1}}} \quad (92)$$

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

The derivatives $\frac{\partial R_i^{x_{k-1},aa}}{\partial \hat{a}_p^{x_k,x_{k-1}}}$, $\frac{\partial R_i^{aa,x_k}}{\partial \hat{a}_a^{x_k,x_{k-1}}}$, $\frac{\partial \hat{a}_a^{x_k,x_{k-1}}}{\partial \vec{a}_a^{x_k,x_{k-1}}}$, and $\frac{\partial R_i^{aa,x_{k-1}}}{\partial \psi_a^{x_k,x_{k-1}}}$ are identical in structure to the derivatives of a rotation matrix with respect to an Euler axis and angle shown in Appendix A.

$$\vec{a}_a = \hat{a}_p \times \hat{a}_c = \begin{bmatrix} \hat{a}_p^{(2)}\hat{a}_c^{(3)} - \hat{a}_p^{(3)}\hat{a}_c^{(2)} \\ \hat{a}_p^{(3)}\hat{a}_c^{(1)} - \hat{a}_p^{(1)}\hat{a}_c^{(3)} \\ \hat{a}_p^{(1)}\hat{a}_c^{(2)} - \hat{a}_p^{(2)}\hat{a}_c^{(1)} \end{bmatrix} \tag{93}$$

$$\frac{\partial \vec{a}_a}{\partial \hat{a}_p} = \begin{bmatrix} 0 & \hat{a}_c^{(3)} & -\hat{a}_c^{(2)} \\ -\hat{a}_c^{(3)} & 0 & \hat{a}_c^{(1)} \\ \hat{a}_c^{(2)} & -\hat{a}_c^{(1)} & 0 \end{bmatrix} \tag{94}$$

$$\psi_a = acos(\hat{a}_p \cdot \hat{a}_c) \tag{95}$$

$$\frac{\partial \psi_a}{\partial \hat{a}_p} = -\frac{1}{\sqrt{1-(\hat{a}_p \cdot \hat{a}_c)^2}}(\hat{a}_c)^T \tag{96}$$

Derivatives with respect to $\hat{a}_c^{x_k,x_{k-1}}$ are similar except that $R_i^{x_{k-1},aa}$ is not dependent on $\hat{a}_c^{x_k,x_{k-1}}$ so the product rule is not required.

$$\frac{\partial R_i^{x_{k-1},x_k}}{\partial \hat{a}_c^{x_k,x_{k-1}}} = (R_i^{x_{k-1},aa} \otimes I)\frac{\partial R_i^{aa,x_k}}{\partial \hat{a}_c^{x_k,x_{k-1}}} \tag{97}$$

$$\frac{\partial R_i^{aa,x_k}}{\partial \hat{a}_c^{x_k,x_{k-1}}} = \frac{\partial R_i^{aa,x_k}}{\partial \hat{a}_a^{x_k,x_{k-1}}}\frac{\partial \hat{a}_a^{x_k,x_{k-1}}}{\partial \vec{a}_a^{x_k,x_{k-1}}}\frac{\partial \vec{a}_a^{x_k,x_{k-1}}}{\partial \hat{a}_c^{x_k,x_{k-1}}} + \frac{\partial R_i^{aa,x_{k-1}}}{\partial \psi_a^{x_k,x_{k-1}}}\frac{\partial \psi_a^{x_k,x_{k-1}}}{\partial \hat{a}_c^{x_k,x_{k-1}}} \tag{98}$$

The derivative $\frac{\partial R_i^{x_{k-1},aa}}{\partial \hat{a}_c^{x_k,x_{k-1}}}$ is identical in structure to the derivatives of a rotation matrix with respect to an Euler axis and angle shown in Appendix A.

$$\frac{\partial \vec{a}_a}{\partial \hat{a}_c} = \begin{bmatrix} 0 & -\hat{a}_p^{(3)} & \hat{a}_p^{(2)} \\ \hat{a}_p^{(3)} & 0 & -\hat{a}_p^{(1)} \\ -\hat{a}_p^{(2)} & \hat{a}_p^{(1)} & 0 \end{bmatrix} \tag{99}$$

$$\frac{\partial \psi_a}{\partial \hat{a}_c} = -\frac{1}{\sqrt{1-(\hat{a}_p \cdot \hat{a}_c)^2}}(\hat{a}_p)^T \tag{100}$$

The derivatives with respect to the articulation angle (ϕ_i) are not dependent on $R_i^{x_{k-1},aa}$ and can be calculated using the 3×3 representation of $\frac{\partial R}{\partial \phi}$ from Appendix A. After multiplication, $\frac{\partial R_i^{x_{k-1},x_k}}{\partial \phi_i}$ must be reshaped to a 9×1 matrix.

$$\frac{\partial R_i^{x_{k-1},x_k}}{\partial \phi_i} = \frac{\partial R_i^{x_{k-1},aa}}{\partial \phi_i} R^{aa,x_k} \tag{101}$$

The rotation also effects the translation. The remaining derivatives required for $\frac{\partial T_i^{w,x_k}}{\partial R_i^{x_{l-1},x_l}}$ are as follows.

$$\frac{\partial T_i^{w,x_k}}{\partial R_i^{w,x_k}} = I \otimes (-J_c^{x_k,x_{k-1}})^T \tag{102}$$

$$\frac{\partial T_i^{w,x_k}}{\partial R_i^{w,x_{k-1}}} = I \otimes (J_p^{x_k,x_{k-1}})^T \tag{103}$$

$$\frac{\partial T_i^{w,x_k}}{\partial T_i^{w,x_{k-1}}} = I \tag{104}$$

Next, the derivatives with respect to the joint locations (J_p and J_c) must be calculated. Since the joint locations do not have an effect on the rotation matrices, only the derivatives with respect to the translations and the joint penalty (B) are required. BB_p and BB_c are the bounding boxes for the parent and the child

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

respectively.

$$\frac{\partial T_i^{w,x_k}}{\partial J_p^{x_{l-1},x_l}} = \begin{cases} R_i^{w,x_{k-1}} & k = l \\ \frac{\partial T_i^{w,x_k}}{\partial T_i^{w,x_{k-1}}} \frac{\partial T_i^{w,x_{k-1}}}{\partial J_p^{x_{l-1},x_l}} & k > l \end{cases} \tag{105}$$

$$\frac{\partial T_i^{w,x_k}}{\partial J_c^{x_{l-1},x_l}} = \begin{cases} -R_i^{w,x_k} & k = l \\ \frac{\partial T_i^{w,x_k}}{\partial T_i^{w,x_{k-1}}} \frac{\partial T_i^{w,x_{k-1}}}{\partial J_c^{x_{l-1},x_l}} & k > l \end{cases} \tag{106}$$

$$d^j = [(J_p^j - BB_p^{max})^T, (BB_p^{min} - J_p^j)^T, (J_c^j - BB_c^{max})^T, (BB_c^{min} - J_c^j)^T] \tag{107}$$

$$\frac{\partial d_k^j}{\partial J_p^j} = [1, 1, 1, -1, -1, -1, 0, 0, 0, 0, 0, 0] \tag{108}$$

$$\frac{\partial d_k^j}{\partial J_c^j} = [0, 0, 0, 0, 0, 0, 1, 1, 1, -1, -1, -1] \tag{109}$$

$$\frac{\partial B_j}{\partial J_p^j} = \lambda_{jp} \sum_{k=1}^{12} \frac{1}{1+e^{d_k \eta}} (\eta e^{d_k \eta}) \frac{\partial d_k^j}{\partial J_p^j} \tag{110}$$

$$\frac{\partial B_j}{\partial J_c^j} = \lambda_{jp} \sum_{k=1}^{12} \frac{1}{1+e^{d_k \eta}} (\eta e^{d_k \eta}) \frac{\partial d_k^j}{\partial J_c^j} \tag{111}$$

Finally, the derivatives with respect to the smoothness constraint (SC) must be calculated.

$$\frac{\partial f}{\partial \phi_i} = \frac{\partial A}{\partial \phi_i} + \frac{\partial SC}{\partial \phi_i} \tag{112}$$

$$\Delta_i = \phi_i - \phi_{i-1} \tag{113}$$

$$\frac{\partial SC}{\partial \phi_i} = \begin{cases} -2\lambda sc(1 - cos(\Delta_{i+1}))sin(\Delta_{i+1}) & \text{first frame} \\ -2\lambda sc\left[(1 - cos(\Delta_{i+1}))sin(\Delta_{i+1}) - (1 - cos(\Delta_i))sin(\Delta_i)\right] & \text{middle frames} \\ 2\lambda_s c(1 - cos(\Delta_i))sin(\Delta_i) & \text{last frames} \end{cases} \tag{114}$$

The derivative of the reprojection error (A) with respect to Ω^{c_j}, R_i^{w,c_j}, and T_i^{w,c_j} are calculated as defined in Appendix A.

REFERENCES

[1] I. Buerk, *SpectralClustering*, 2012. [Online]. Available: `https://www.mathworks.com/matlabcentral/fileexchange/34412-fast-and-efficient-spectral-clustering`.

[2] D. H. Curtis and R. G. Cobb, "Satellite Articulation Sensing using Computer Vision," in *Proceedings of AIAA SciTech 2017*, 2017.

[3] J. D'Errico, *Inhull*, 2006. [Online]. Available: `https://www.mathworks.com/matlabcentral/fileexchange/10226-inhull`.

[4] G. Fasano, M. Grassi, and D. Accardo, "A stereo-vision Based System for Autonomous Navigation of an In-Orbit Servicing Platform," in *AIAA Infotech@ Aerospace*, Seattle, Washington, 2009, pp. 1–10.

[5] J. Fayad, C. Russell, and L. Agapito, "Automated articulated structure and 3D shape recovery from point correspondences," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 431–438, 2011.

[6] V. Ghadiok, J. Goldin, and D. Geller, "Gyro-Aided Vision-Based Relative Pose Estimation for Autonomous Rendezvous and Docking," *Advances in the Astronautical Sciences*, vol. 149, pp. 713–728, 2013.

[7] S. J. Hannah, "A relative navigation application of ULTOR technology for automated rendezvous and docking," in *Proceedings of SPIE, Spaceborne Sensors III*, R. T. Howard and R. D. Richards, Eds., vol. 6220, 2006, pp. 1–12.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

[8] C.-C. J. Ho and N. H. McClamroch, "Automatic spacecraft docking using computer vision-based guidance and control techniques," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 2, pp. 281–288, 1993.

[9] R. T. Howard, A. F. Heaton, R. M. Pinson, and C. K. Carrington, "Orbital Express Advanced Video Guidance Sensor," in *IEEE Aerospace Conference Proceedings*, 2008, pp. 1–10.

[10] S. J. Kelly, "A Monocular SLAM method to Estimate Relative Pose During Satellite Proximity Operations," Thesis, Air Force Institute of Technology, 2015, p. 118.

[11] U. V. Luxburg, "A Tutorial on Spectral Clustering A Tutorial on Spectral Clustering," *Statistics and Computing*, vol. 17, no. March, pp. 395–416, 2006.

[12] M. Marques and J. Costeira, "Optimal shape from motion estimation with missing and degenerate data," in *2008 IEEE Workshop on Motion and Video Computing, WMVC*, 2008, pp. 2–7.

[13] B. J. Naasz, J. Van Eepoel, S. Z. Queen, C. M. Southward, and J. Hannah, "Flight results from the HST SM4 Relative Navigation Sensor system," in *33rd Annual AAS Guidance and Control Conference*, Breckenridge, Colorado, 2010.

[14] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems*, pp. 849–856, 2002.

[15] M. Paladini, A. Del Bue, J. Xavier, L. Agapito, M. Stošić, and M. Dodig, "Optimal metric projections for deformable and articulated structure-from-motion," *International Journal of Computer Vision*, vol. 96, no. 2, pp. 252–276, 2012.

[16] N. K. Philip and M. R. Ananthasayanam, "Relative position and attitude estimation and control schemes for the final phase of an autonomous docking mission of spacecraft," *Acta Astronautica*, vol. 52, pp. 511–522, 2003.

[17] S. Pillai, M. R. Walter, and S. Teller, "Learning Articulated Motions From Visual Demonstration," in *Robotics: Science and Systems*, 2014.

[18] C. Russell, R. Yu, and L. Agapito, "Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes," in *European conference on computer vision*, Springer International Publishing, 2014, pp. 583–598.

[19] C Sabol, R Burns, and C. A. McLaughlin, "Satellite formation flying design and evolution," *Journal of Spacecraft and Rockets*, vol. 38, no. 2, pp. 270–278, 2001.

[20] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[21] C Tomasi and T Kanade, "Shape and motion from image streams under orthography: a factorization method.," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[22] P. Tresadern and I. Reid, "Articulated structure from motion by factorization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 1110–1115.

[23] B. E. Tweddle and A. Saenz-Otero, "Relative Computer Vision-Based Navigation for Small Inspection Spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 5, pp. 969–978, 2015.

[24] B. E. Tweddle and D. W. Miller, "Computer Vision-Based Localization and Mapping of an Unkown, Uncooperative and Spinning Target for Spacecraft Proximity Operations," Dissertation, Massachusetts Institute of Technology, 2013, p. 410.

[25] R. Vidal, "Subspace Clustering," *IEEE Signal Processing Magazine*, no. March, pp. 52–68, 2011.

[26] J. Yan and M. Pollefeys, "A factorization-based approach to articulated motion recovery," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 815–821.

[27] —, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *European conference on computer vision*, Springer Berlin Heidelberg, 2006, pp. 94–106.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

[28] F. Yu, Z. He, B. Qiao, and X. Yu, "Stereo-vision-based relative pose estimation for the rendezvous and docking of noncooperative satellites," *Mathematical Problems in Engineering*, vol. 2014, p. 12, 2014.

[29] K. Yucer, O. Wang, A. Sorkine-Hornung, and O. Sorkine-Hornung, "Reconstruction of Articulated Objects from a Moving Camera," in *IEEE International Conference on Computer Vision Workshops*, 2015, pp. 28–36.

[30] P. B. Zhang and Y. S. Hung, "Articulated Structure from Motion through Ellipsoid Fitting," in *International Conference of Image Processing, Computer Vision and Pattern Recognition*, 2015, pp. 179–186.

Copyright © 2017 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com